



Faculté des Sciences
Département de Géographie

Intégration de la continuité spatiale dans la structure multidimensionnelle d'un entrepôt de données

SOLAP raster

par
Jean-Paul Kasprzyk

Thèse soutenue le 1^{er} juillet 2015
en vue de l'obtention du titre de
Docteur en Sciences

Membres du jury

Prof. Roland Billen (secrétaire)	Université de Liège
Prof. Sandro Bimonte	IRSTEA de Clermont-Ferrand (France)
Prof. Yves Cornet (président)	Université de Liège
Prof. Jean-Paul Donnay (promoteur)	Université de Liège
Prof. Thérèse Libourel	UM Université de Montpellier (France)
Prof. Jef Wijsen	Université de Mons
Prof. Pierre Wolper	Université de Liège

Résumé

Les progrès technologiques de ces dernières décennies ont engendré une acquisition massive de données numériques dont le volume croît exponentiellement. Afin de retirer efficacement l'information qu'elles contiennent, de puissants outils de collecte, de stockage et d'analyse ont été développés et rassemblés dans une discipline appelée « informatique décisionnelle ». Parmi ces outils, les entrepôts de données sont responsables de l'archivage des données en les structurant de manière multidimensionnelle (temps, espace ou autres). On parle alors d'hypercubes de données ou de cubes de données lorsqu'ils sont limités à trois dimensions. Ces hypercubes peuvent alimenter des systèmes OLAP (« On Line Analytical Processing ») dont le rôle est de synthétiser rapidement l'information sous forme de tableaux et graphiques interactifs destinés aux décideurs de divers domaines : marketing, environnement, criminologie, *etc.* L'utilisateur navigue ainsi dans un hypercube au moyen d'opérations OLAP telles que des coupes sur les membres de dimension (par exemple : agrégation des données sur le mois de janvier de la dimension temporelle) ou des forages dans des hiérarchies (par exemple : passage du niveau « année » au niveau « mois » de la dimension temporelle). Lorsque l'OLAP est couplé aux techniques d'analyse spatiale proposées par les systèmes d'information géographique (SIG), une interface cartographique vient enrichir l'exploration de l'utilisateur et les opérations OLAP peuvent être appliquées sur des dimensions définies dans l'espace géographique (forage spatial ou coupe spatiale). Ce type d'outil est appelé SOLAP (« Spatial OLAP »).

Les SOLAP proposés actuellement sur le marché souffrent tous de la même carence : ils sont incapables de représenter les dimensions spatiales (X , Y) de manière continue. Cette représentation est pourtant essentielle pour la gestion des phénomènes spatialement continus (température, pollution, *etc.*), mais aussi pour visualiser des phénomènes spatialement discrets (ventes de produits, crimes, *etc.*) tout en minimisant le problème des unités spatiales modifiables (« Modifiable Areal Unit Problem » ou MAUP). Ce type de visualisation est notamment utilisé par la police pour prédire l'emplacement des futurs délits au moyen de cartes de points chauds (« hotspots ») en estimant leur densité par la méthode des noyaux (« Kernel Density Estimation » ou KDE). Dans le domaine du SIG, les données raster (en opposition aux données vectorielles) permettent une représentation efficace de la continuité spatiale à travers des grilles numériques géoréférencées. Alors que les outils SOLAP actuels ne considèrent que le modèle vectoriel, cette recherche exploite le modèle raster afin d'intégrer la continuité spatiale dans la structure multidimensionnelle d'un entrepôt de données alimentant un SOLAP (« SOLAP raster »).

Malgré sa sous-exploitation dans la littérature SOLAP, le raster présente de nombreux points communs avec un type de cube de données particulier : le cube MOLAP (« Multidimensional OLAP »). A l'instar d'une image satellitaire (raster) représentant les deux dimensions spatiales planimétriques et une dimension « bande spectrale », un cube MOLAP est un tableau tridimensionnel dont les coordonnées des cellules (comparables aux pixels d'un raster) permettent une indexation efficace des membres de dimensions décrivant les faits analysés par l'utilisateur. Dans un premier modèle que nous appelons « cube raster », nous définissons les bases d'un SOLAP raster à trois dimensions en partant de la définition d'un cube MOLAP. Contrairement aux SOLAP vectoriels où la spatialité

est rattachée à une dimension sémantique au moyen de pointeurs vers des géométries, notre modèle intègre directement les dimensions spatiales planimétriques (X, Y) dans la structure multidimensionnelle de l'entrepôt de données. Grâce à cette caractéristique originale, n'importe quelle entité géographique (pays, bâtiment, route, *etc.*) peut être importée à la volée comme membre dans l'analyse de l'utilisateur, ce qui est difficilement envisageable avec un outil SOLAP vectoriel classique. Une extension de ce modèle SOLAP, appelée « hypercube raster », est ensuite développée en confiant la gestion de dimensions non-spatiales supplémentaires à un système de gestion de base de données (SGBD) relationnel (« Relational OLAP » ou ROLAP). L'hypercube raster est ensuite peuplé par des champs raster KDE représentant des densités de criminalité en fonction d'un espace continu (dimensions raster), des types de délits, et du temps (dimensions ROLAP). Notre modèle est ainsi capable de combiner la production de cartes de hotspots à différentes échelles d'analyse avec les opérations de navigation SOLAP : forages dans la hiérarchie d'une dimension spatiale ou non-spatiale, et coupes selon un ou plusieurs membres spatiaux ou non-spatiaux.

Le modèle d'hypercube raster exploitant des champs KDE est validé par un prototype opérationnel basé sur une architecture complètement « open source ». Plusieurs jeux de données y sont intégrés, notamment des données de criminalité provenant de Londres et Seattle. A la fin de ce travail, les résultats d'une étude comparative entre SOLAP raster et SOLAP vectoriel démontrent qu'une architecture SOLAP hybride vecteur/raster présenterait le même intérêt d'un point de vue spatial que les architectures hybrides ROLAP/MOLAP pour des données purement sémantiques (gestion des hypercubes détaillés par le vecteur ou le ROLAP et gestion des hypercubes moins détaillés par le raster ou le MOLAP).

Mots-clés : SOLAP, raster, informatique décisionnelle, entrepôt de données, cube de données, SIG, cartographie criminelle, champ continu.

Abstract

Technological advances in recent decades have created a massive acquisition of digital data whose volume grows exponentially. To efficiently extract the information they contain, powerful tools have been developed to collect, store and analyze these data. These tools are gathered in a discipline called “business intelligence”. Among them, data warehouses are responsible for archiving data by structuring them in a multidimensional way (time, space or others). They are called data hypercubes or data cubes when they are limited to three dimensions. Hypercubes can supply OLAP (On Line Analytical Processing) systems that aim at quickly synthesizing information in interactive tables and charts for decision-makers from various fields: marketing, environment, criminology, etc. Thus, users can navigate into hypercubes using OLAP operations such as slicing on dimension members (e.g. data aggregation for the month of January in the time dimension), or drilling into hierarchies (e.g. switching from the “year” level to the “month” level in the time dimension). When OLAP is coupled with spatial analysis techniques supplied by geographic information systems (GIS), a map interface then improves the exploration of data: OLAP operations can be applied to dimensions defined in the geographical space (spatial drilling or spatial slicing). This kind of tool is called SOLAP (Spatial OLAP).

SOLAP tools currently available on the market all suffer from the same deficiency: they are unable to represent spatial dimensions (X , Y) in a continuous way. This representation is nevertheless essential for the management of spatially continuous phenomena (temperature, pollution, etc.) but also for visualizing spatially discrete events (product sales, crimes, etc.) while minimizing the Modifiable Areal Unit Problem (MAUP). This kind of visualization is used especially by the police to predict the location of future crimes through hotspot maps which are generated by the Kernel Density Estimation (KDE) method. In the field of GIS, raster data (as opposed to vector data) enable effective representation of spatial continuity through digital georeferenced grids. Whereas current SOLAP tools only consider vector data, our research uses the raster model to integrate spatial continuity into the multidimensional structure of a data warehouse feeding a SOLAP (“raster SOLAP”).

Despite its underutilization in the SOLAP literature, the raster model has many similarities with a particular kind of data cube: the MOLAP cube (Multidimensional OLAP). Like a satellite image (raster) representing the two planimetric spatial dimensions and one “spectral band” dimension, a MOLAP cube is a three dimensional array whose cells’ coordinates (similar to raster pixels) enable an efficient indexation of dimensions’ members (describing the analyzed facts). In a first original model that we call “raster cube” we define the bases for a three-dimensional raster SOLAP, starting from the definition of a MOLAP cube. Unlike vector SOLAP - where spatiality is attached to a semantic dimension through pointers to geometries - our model directly integrates spatial dimensions (X , Y) in the multidimensional structure of the data warehouse. With this original feature, any geographical entity (country, building, road, etc.) can be imported on the fly as a member in the analysis of the user, which is hardly possible with conventional vector SOLAP tools. An extension of this SOLAP model, called “raster hypercube”, is then developed by entrusting the management of extra non-spatial dimensions to a relational

database management system (Relational OLAP or ROLAP). The raster hypercube is then populated by KDE raster fields representing crime densities, which are defined in a continuous space (raster dimensions) through time and crime types (ROLAP dimensions). Our model is able to combine the production of hotspot maps at different scales of analysis with SOLAP navigation operations: slicing on spatial or non-spatial members, and drilling into the hierarchy of spatial or non-spatial dimensions.

Our raster hypercube model is validated by an operating prototype which is based on open source tools only. Several datasets are integrated through KDE fields, including crime data from London and Seattle. At the end of our work, the results of a comparative study between raster SOLAP and vector SOLAP demonstrate that hybrid vector/raster SOLAP architectures present the same interest for spatial data as hybrid ROLAP/MOLAP architectures do for purely semantic data (management of detailed hypercubes by vector or ROLAP and management of less detailed hypercubes by raster or MOLAP).

Keywords : SOLAP, raster, business intelligence, data warehouse, data cube, GIS, crime mapping, continuous field.

Table des matières

Résumé.....	i
Abstract.....	iii
Table des matières.....	v
Liste des tableaux.....	ix
Liste des figures	xi
Remerciements.....	xvii

Introduction.....	1
Question de recherche.....	1
Plan de la thèse.....	3

Partie 1: état de l'art

Chapitre 1 : informatique décisionnelle	7
1.1. ETL	8
1.2. Entrepôts de données	9
1.2.1. Dimensions, membres, hiérarchies et niveaux.....	10
1.2.2. Faits et mesures.....	13
1.2.3. Modélisation conceptuelle	14
1.2.4. Modélisation logique	15
1.2.5. Cubes et cuboïdes de données.....	16
1.3. OLAP	18
1.3.1. Historique et caractéristiques	18
1.3.2. Opérations OLAP.....	19
1.3.3. Interfaces OLAP	20
1.3.4. Le langage MDX.....	22
1.3.5. Architectures OLAP	22
1.3. Conclusion	26

Chapitre 2 : SIG et format raster.....	27
2.1. Dualités de l'information spatiale.....	28
2.1.1. Dualité des phénomènes géographiques : discrets et continus	28
2.1.2. Dualité de la modélisation spatiale : vecteur et raster.....	29
2.1.3. Bilan.....	33
2.2. Couverture spatiale	35
2.3. Caractéristiques du format raster	36
2.3.1. Algèbre de carte	36
2.3.2. Optimisation des traitements raster.....	37
2.4. Modèles physiques de stockage raster	40
2.4.1. Oracle GeoRaster	40
2.4.2. PostGIS Raster	43
2.5. Conclusion	44

Chapitre 3 : SOLAP	47
3.1. Spatialisation de l'entrepôt de données.....	48
3.1.1. Dimension spatiale.....	48

3.1.2.	Mesure spatiale	49
3.2.	Interface cartographique	50
3.3.	Opérations SOLAP	51
3.4.	Architectures SOLAP	52
3.5.	Classification des outils SOLAP.....	52
3.5.1.	OLAP dominant.....	53
3.5.2.	SIG dominant.....	53
3.5.3.	OLAP-SIG intégré	54
3.5.4.	Bilan.....	56
3.6.	Conclusion	57
Chapitre 4 : hypothèse		59
4.1.	SOLAP raster.....	60
4.1.1.	Modèles SOLAP maillés.....	60
4.1.2.	Optimisation des requêtes raster	65
4.2.	SOLAP continu.....	65
4.2.1.	Ahmed & Miquel (2005)	66
4.2.2.	Vaisman & Zimányi (2009).....	66
4.2.3.	Gomez <i>et al</i> (2010)	67
4.2.4.	Bimonte & Kang (2010)	68
4.2.5.	Gomez <i>et al</i> (2012)	69
4.2.6.	Zaamoune <i>et al</i> (2013).....	69
4.2.7.	Comparaison	69
4.3.	Conclusion : hypothèse de recherche.....	72

Partie 2: modélisation SOLAP raster

Chapitre 5 : modèle raster multidimensionnel		75
5.1.	Introduction.....	75
5.2.	Cube raster	76
5.3.	Dimensions raster spatiales (R , C).....	80
5.3.1.	Membres spatiaux	80
5.3.2.	Hiérarchies spatiales	81
5.4.	Troisième Dimension raster (B).....	87
5.4.1.	Membres et hiérarchies d'une dimension quelconque.....	88
5.4.2.	Membres et hiérarchies d'une dimension continue	89
5.5.	Membres spatialisés (s).....	92
5.5.1.	Raster dimensionnel intégré.....	93
5.5.2.	Conditions sur les métadonnées spatiales d'un raster dimensionnel	94
5.5.3.	Intégration à la volée d'un raster dimensionnel	97
5.5.4.	Intégration à la volée d'une couche vectorielle	100
5.6.	Exemple de cube raster	100
5.7.	Opérations élémentaires.....	102
5.7.1.	Filtrage vertical	103
5.7.2.	Filtrage horizontal.....	104
5.7.3.	Agrégation partielle	107
5.7.4.	Agrégation totale.....	110
5.8.	Interfaces SOLAP raster	112
5.8.1.	La carte remplace la table de pivot	112

5.8.2.	Interface cartographique et dimensions spatiales	113
5.8.3.	Interface cartographique et membres spatialisés	114
5.8.4.	Interface graphique et dimension non-spatiale	116
5.8.5.	Interface graphique et membres spatialisés	117
5.9.	Opérations SOLAP	119
5.9.1.	Opérations de coupe.....	119
5.9.2.	Opérations de forage	123
5.9.3.	Opérations de comparaison.....	126
5.10.	Entrelacement	128
5.11.	Conclusion	130
Chapitre 6 : modèle raster relationnel		133
6.1.	Introduction.....	133
6.2.	Hypercube raster	134
6.3.	Structure multidimensionnelle	136
6.3.1.	Fait et mesure	136
6.3.2.	Dimensions, membres, hiérarchies et cuboïdes	137
6.3.3.	Membres spatialisés	139
6.4.	Métadonnées des mesures ROLAP.....	139
6.5.	Exemple d'hypercube raster	141
6.6.	Opérations ROLAP élémentaires.....	142
6.6.1.	Filtrage vertical ROLAP	144
6.6.2.	Filtrage horizontal ROLAP	145
6.6.3.	Agrégations ROLAP	145
6.6.4.	Ordre des opérations élémentaires	147
6.7.	Opérations SOLAP et interfaces du modèle ROLAP	148
6.8.	Intérêt de la dimension <i>B</i> dans le modèle ROLAP	150
6.9.	Conclusion	151
Chapitre 7 : modèle raster continu		153
7.1.	Introduction : hypercube raster continu	153
7.2.	Interpolation des champs	154
7.3.	Estimation de la densité par la méthode des noyaux (KDE)	156
7.3.1.	Champ KDE.....	156
7.3.2.	Influence de la fenêtre de lissage et de la fonction KDE	158
7.3.3.	Détermination des hotspots de criminalité.....	159
7.4.	Intégration de champs KDE dans un hypercube raster.....	160
7.4.1.	Hypothèse SOLAP KDE.....	161
7.4.2.	Démonstration de l'hypothèse SOLAP KDE	162
7.4.3.	Conséquence de l'hypothèse SOLAP KDE	163
7.4.4.	Démonstration de la conséquence de l'hypothèse SOLAP KDE	163
7.4.5.	Métadonnées des mesures ROLAP KDE	164
7.5.	Exemple d'hypercube raster KDE	165
7.6.	Opérations SOLAP sur champs raster KDE	166
7.6.1.	Interface cartographique	167
7.6.2.	Interface graphique	174
7.7.	Conclusion	175

Partie 3: validation et conclusion

Chapitre 8 : validation.....	181
8.1. Introduction.....	181
8.2. Cas d'étude.....	181
8.2.1. Hypercube London.....	182
8.2.2. Hypercube Seattle	183
8.2.3. Cube Moon.....	184
8.3. Prototype SOLAP raster	186
8.3.1. Architecture globale.....	186
8.3.2. Entrepôt de données	187
8.3.3. Serveur SOLAP	193
8.3.4. Interface client	197
8.4. SOLAP raster et agrégations holistiques	207
8.5. Étude comparative	209
8.5.1. Présentation des trois modèles comparés.....	209
8.5.2. Comparaison	211
8.5.3. Bilan et discussion	214
8.6. Perspectives.....	217
8.6.1. Hypercube raster tuilé	217
8.6.2. « Network OLAP »	220
8.6.3. Cube raster dimensionnel.....	221
8.7. Conclusion	223
Conclusion générale.....	225
Synthèse des différentes parties	225
Contribution au domaine SOLAP	234
Contribution au domaine « crime mapping »	235
Perspectives de recherche	236
Bibliographie.....	239
 Annexes	
Annexe 1. Fichiers XML de définition des hypercubes raster.....	iii
Annexe 2. Exemple de fichier map pour la définition WMS de l'interface "raster cube"	xiii
Annexe 3. Fonction KDE dans PostGIS	xv
Annexe 4. Requêtes SQL de l'étude comparative	xix

Liste des tableaux

Tableau 1. Différentes caractéristiques de hiérarchies dimensionnelles	12
Tableau 2. Comparaison des différents modèles SOLAP intégrant la continuité spatiale ..	71
Tableau 3. Comparaison du volume de données des trois hypercubes raster	182
Tableau 4. Temps de réponse de l'interface cartographique (SOLAP raster/vecteur)	212
Tableau 5. Temps de réponse de l'interface graphique (SOLAP raster/vecteur)	212

Liste des figures

Figure 1. Architecture décisionnelle	7
Figure 2. Exemple de traitement ETL.....	8
Figure 3. Exemple de dimension temporelle hiérarchisée	11
Figure 4. Schéma en étoile.....	15
Figure 5. Schéma en flocon	16
Figure 6. Treillis de cuboïdes.....	17
Figure 7. Opération « drill down » dans une table de pivot.....	21
Figure 8. Graphique issu d'un cube de données	21
Figure 9. Architecture ROLAP	23
Figure 10. Architecture MOLAP	24
Figure 11. Agrégation dans un cube MOLAP	24
Figure 12. Architecture HOLAP.....	26
Figure 13. Deux stratégies de modélisation d'un champ	29
Figure 14. Les trois primitives vectorielles.....	30
Figure 15. La matrice des 9 intersections	30
Figure 16. Exemple de raster de dimensions 3*3*3	32
Figure 17. Les différents découpages spatiaux héritant de la classe « coverage » (OGC) ..	35
Figure 18. Les trois types de fonctions d'algèbre de carte.....	36
Figure 19. Les différents tuilages raster proposés dans PostGIS.....	39
Figure 20. Exemple de pyramide raster à trois niveaux.....	40
Figure 21. Stockage physique d'un raster dans Oracle GeoRaster	41
Figure 22. Stockage physique d'un raster dans PostGIS	43
Figure 23. Les trois types de dimensions spatiales d'un outil SOLAP.....	49
Figure 24. Exemple d'intégration de graphiques dans le SOLAP Map4Decision.	50
Figure 25. Exemple d'opération « drill down » avec l'interface de Map4Decision.....	51
Figure 26. Exemple d'application OLAP-dominant.	53
Figure 27. Exemple d'application SIG-dominant.	54
Figure 28. Les trois types de hiérarchies d'une dimension géographique	55
Figure 29. Caractéristiques de l'OLAP et du SIG présentes dans le SOLAP.....	59
Figure 30. Schéma en constellation pour les inventaires forestiers	61
Figure 31. Les 7 types de dimensions spatiales en distinguant le vecteur et le matriciel ...	62
Figure 32. Les deux types de cubes de données intégrant la structure matricielle	62
Figure 33. Scénarios d'intégration des couvertures dans les cubes matriciels	63
Figure 34. Élaboration d'un cube de données continu.....	66
Figure 35. Exemple de dimensions et de mesure de type « champ continu »	67
Figure 36. Hiérarchie d'un objet géographique « champ continu »	68
Figure 37. Schéma conceptuel à trois dimensions <i>B</i> , <i>R</i> et <i>C</i>	76
Figure 38. Position d'une mesure atomique dans l'espace d'un cube de données	77
Figure 39. Transformation de l'espace raster vers l'espace géographique	78
Figure 40. Différence entre un cube raster et un cube MOLAP tridimensionnel	79
Figure 41. Schéma conceptuel d'un cube raster	80
Figure 42. Représentations d'entités spatiales par combinaisons logiques de <i>R</i> et <i>C</i>	81
Figure 43. Pyramide raster à 3 niveaux	83

Figure 44. Hiérarchie SOLAP implicite des dimensions spatiales <i>X</i> et <i>Y</i>	83
Figure 45. Relation implicite entre membres spatiaux	85
Figure 46. Exemple de hiérarchie spatiale matérialisée par des cuboïdes	86
Figure 47. Exemple de hiérarchisation de la dimension <i>B</i>	88
Figure 48. Relation directe entre un membre spatialisé et le fait d'un cube raster.....	94
Figure 49. Exemples de relations entre un cube raster et un raster dimensionnel	96
Figure 50. Découpage d'un cube raster et d'un raster dimensionnel.....	99
Figure 51. Exemple de cube raster (schéma en étoile)	101
Figure 52. Chaîne d'opérations appliquée sur un entrepôt raster multidimensionnel	103
Figure 53. Les deux types de filtrages horizontaux d'un cube raster	104
Figure 54. Redéfinition des pointeurs d'une dimension <i>B</i> après un filtrage.....	105
Figure 55. Application d'un filtre spatial à partir de membres spatialisés	107
Figure 56. Agrégation partielle non-spatiale d'un cube raster.....	108
Figure 57. Agrégations totales non-spatiale et spatiale d'un cube raster.....	111
Figure 58. Représentation des dimensions spatiales dans l'interface cartographique	114
Figure 59. Représentation de membres spatialisés dans l'interface cartographique	115
Figure 60. Représentation d'une dimension non-spatiale dans l'interface graphique	116
Figure 61. Interface graphique affichant une dimension <i>B</i> et trois membres spatialisés...	118
Figure 62. Résultat d'une coupe non-spatiale.....	120
Figure 63. Création d'un membre spatialisé et résultat d'une coupe spatiale	122
Figure 64. Forage non-spatial	124
Figure 65. Forage spatial.....	125
Figure 66. Comparaison entre deux coupes d'un cube raster	127
Figure 67. Les trois types d'entrelacements d'un cube raster.....	129
Figure 68. Schéma conceptuel d'un hypercube raster	135
Figure 69. Schéma logique d'un hypercube raster.....	135
Figure 70. Différentes combinaisons de faits ROLAP et raster pour un fait global	137
Figure 71. Schéma logique d'un hypercube raster comptant 3 niveaux spatiaux.....	138
Figure 72. Table des métadonnées d'un hypercube raster	140
Figure 73. Espace de stockage occupé par un hypercube raster	141
Figure 74. Exemple de schéma conceptuel d'un hypercube raster	142
Figure 75. Chaîne d'opérations appliquée sur un entrepôt raster relationnel	144
Figure 76. Opération locale d'algèbre de carte (conventionnel et cubique).....	146
Figure 77. Interface graphique affichant une dimension <i>B</i> et une dimension ROLAP	149
Figure 78. Illustration du problème des unités spatiales modifiables.....	156
Figure 79. Exemples de fonctions KDE	159
Figure 80. Construction de hotspots par KDE	160
Figure 81. Table des métadonnées d'un hypercube raster KDE.....	165
Figure 82. Exemple de schéma conceptuel d'un hypercube raster KDE.....	166
Figure 83. Opérations de coupe dans un hypercube raster KDE	169
Figure 84. Opération de comparaison cartographique dans un hypercube raster KDE	170
Figure 85. Opération de coupe spatiale dans un hypercube raster KDE	171
Figure 86. Opération de forage spatial dans un hypercube raster KDE.....	173
Figure 87. Comparaison entre les mesures KDE et « nombre de délits »	175
Figure 88. Schéma conceptuel de l'hypercube raster « London ».....	183
Figure 89. Schéma conceptuel de l'hypercube raster « Seattle »	184
Figure 90. Schéma conceptuel du cube raster « Moon »	185
Figure 91. Exemple de coupe temporelle sur le cube « Moon ».....	185
Figure 92. Architecture du prototype SOLAP raster	187

Figure 93. Tables principales de l'entrepôt de données SOLAP raster	189
Figure 94. Extrait de la vue de métadonnées « raster_columns ».....	190
Figure 95. Exemples de requêtes SOLAP SQL dans l'interface cartographique	192
Figure 96. Exemple de requête SOLAP SQL dans l'interface graphique	193
Figure 97. Architecture SOLAP hybride vecteur/raster (KDE).....	195
Figure 98. Création d'un hypercube raster KDE	197
Figure 99. Aperçu général de l'interface client « Raster Data Cube »	198
Figure 100. Paramétrage d'une opération SOLAP	200
Figure 101. Gestionnaire de couches OpenLayers	201
Figure 102. Paramétrage de classification et légende	202
Figure 103. Digitalisation de deux membres spatialisés.....	203
Figure 104. Coupe spatiale selon les deux geomembres digitalisés	204
Figure 105. Interface graphique : 2 dimensions non-spatiales et 2 membres spatialisés ..	206
Figure 106. Mise en évidence d'un hotspot avec la fonction « range ».....	208
Figure 107. Approches SOLAP raster, SOLAP vecteur et SIG : hypercube London	210
Figure 108. Digitalisation des 4 geomembres et résultat de la requête 12	213
Figure 109. Gestion de la faible densité des dimensions raster dans un raster tuilé.....	217
Figure 110. Exemple de schéma en constellation pour un hypercube raster tuilé.....	219
Figure 111. Exemple de requête SOLAP distribuée vers un « cloud »	221
Figure 112. Fonction zonale d'algèbre de carte conventionnelle et cubique.....	222

*A Joséphine Martin, plus qu'une grand-
mère, un modèle*

Remerciements

Mes premiers remerciements sont naturellement adressés à mon promoteur, le professeur Jean-Paul Donnay, pour avoir cru en moi et pour m'avoir permis de mener à bien cette recherche, sans pour autant abandonner ma passion pour la musique. D'un point de vue scientifique comme d'un point de vue humain, j'ai toujours été parfaitement encadré durant ces années de doctorat et d'assistanat. Merci pour tout!

Je tiens également à remercier les professeurs Thérèse Libourel et Jef Wijzen qui, en tant que membres du comité, ont suivi la thèse depuis le début. Leur disponibilité et leurs conseils pertinents m'ont permis d'avancer vers la bonne voie.

Ce travail est né d'une collaboration avec la Police Fédérale, et cette collaboration n'aurait pas été possible sans l'appui du Commissaire Marc Simon, également membre du comité. Grâce à son ouverture vers le monde académique, une véritable cellule de recherche en « crime mapping » a pu voir le jour à l'Unité de Géomatique. Merci.

Je remercie également les professeurs Sandro Bimonte, Thierry Badard et Yvan Bédard pour nos échanges dans le domaine du SOLAP. Leurs avis d'expert m'ont permis d'orienter ma recherche vers des pistes solides et passionnantes.

Merci aux professeurs Pierre Wolper, Yves Cornet et Roland Billen pour avoir accepté de consacrer leur temps à ce travail en tant que membres du jury.

Merci à Nadia, ma collègue de bureau avec qui nous avons partagé de nombreux moments de réflexion, et dont la naturelle sympathie m'a permis de travailler dans une ambiance des plus agréables. Cette atmosphère est aussi le fruit de tous les collègues de l'Unité de Géomatique, du Département de Géographie et de l'UNIPC que je remercie chaleureusement : Marc, Marie, Gilles, Pierre, Matthieu, Benoît, Luc, ... En particulier, toute ma gratitude va à Robert Sip pour son aide précieuse dans la réalisation de mon prototype, mais aussi pour les « jam sessions » après le boulot.

Merci à mes camarades musiciens qui ont fait preuve de compréhension et de patience par rapport à ma relative indisponibilité lors des six derniers mois de rédaction de ce manuscrit.

Merci à mes amis de toujours, Hyun, Arnaud, Mike, ..., et merci à toute ma famille pour leurs encouragements et leur soutien, autant dans les moments agréables que difficiles. En particulier, merci à mes parents et à ma petite sœur Joëlle pour la relecture de la thèse durant la dernière ligne droite. Merci à Jean Van de Vyvere pour la relecture des articles en anglais.

Last but not least, merci à toi Laura. En plus du SOLAP raster, cette recherche m'aura permis de trouver la femme avec qui j'espère partager le reste de ma vie...

Introduction

Question de recherche

Trois mille. C'est le nombre de délits répertoriés quotidiennement dans la ville de Londres¹. A l'échelle nationale, des dizaines de millions de données sont ainsi produites chaque année par les services de police. Parallèlement, les citoyens eux-mêmes génèrent une quantité astronomique de données numériques de tous types : transactions bancaires, appels téléphoniques, e-mails, activités sur les réseaux sociaux, *etc.* Ces importantes sources d'information sont précieuses pour les experts en charge de la lutte contre la criminalité. Néanmoins, elles nécessitent de puissants outils informatiques de collecte, d'archivage et de synthétisation afin d'être exploitées efficacement.

Suite aux attentats du 11 septembre 2001, le FBI entama la conception de l'IDW (« Investigative Data Warehouse »). Cet entrepôt de données a précisément pour but de centraliser toute l'information susceptible d'aider à prévenir la criminalité, en particulier le terrorisme (Mena, 2003). Mais dans une architecture d'informatique décisionnelle², l'entrepôt à lui seul ne suffit pas. L'analyse des données est possible de manière automatique grâce à des algorithmes relevant de la fouille de données (« data mining »), ou manuellement grâce aux systèmes OLAP (« Online Analytical Processing »).

L'OLAP, discipline de l'informatique décisionnelle apparue au début des années nonante (Codd, 1993), offre une analyse des archives stockées dans un entrepôt (ou cube de données) de manière rapide, conviviale et synthétique. Côté utilisateur, l'OLAP propose des tableaux et des graphiques interactifs. Ces interfaces représentent des agrégats de données, appelées « mesures », définis selon plusieurs dimensions telles que le temps, le type de délit, *etc.* Bien qu'initialement, les outils OLAP fussent principalement destinés aux décideurs d'entreprises commerciales, leur champ d'application s'est progressivement étendu vers divers domaines scientifiques tels que l'agriculture, la sismologie, l'environnement, et bien sûr la criminologie.

Vers la fin des années nonante, les recherches menées à l'Université Laval (Bédard, 1997) donnèrent naissance à l'OLAP spatial (SOLAP). Cette discipline prometteuse intègre la composante spatiale des données dans l'OLAP en s'inspirant des systèmes d'information géographique (SIG). De cette manière, l'utilisateur analyse ses archives de données à travers des tableaux, des graphiques, mais aussi des cartes interactives. Aujourd'hui encore, le SOLAP alimente de nombreuses recherches en géomatique, notamment celle présentée dans ce travail.

¹ Moyenne sur l'année 2012 d'après les données de la « City of London Police » et de la « Metropolitan Police (UK Police, 2014).

² Discipline apparue à la fin du 20^{ème} siècle dont le but est de fournir des outils capables de collecter, stocker et analyser de très grands jeux de données. Elle se traduit en anglais par le terme « Business Intelligence ».

Cette thèse résulte d'un partenariat entre l'Unité de Géomatique de l'Université de Liège et les analystes criminels de la Police Fédérale belge. Dans un premier temps, l'objectif visé était l'intégration des données de criminalité dans un SOLAP afin d'offrir un outil adapté aux besoins des analystes stratégiques³. Plusieurs échanges ont ainsi eu lieu avec les experts de la police belge, mais aussi avec des chercheurs en cartographie criminelle à un niveau international. Un atelier très enrichissant a d'ailleurs été organisé autour du thème de l'analyse criminelle lors de la conférence SAGEO (« Spatial Analysis and Geomatics ») qui s'est tenue à Liège au mois de novembre 2012.

Suite à ces échanges et à diverses expérimentations, nous avons constaté que les outils SOLAP du marché étaient inadaptés à certaines méthodes d'analyse spatiale couramment employées par la police. L'une d'entre elles a particulièrement retenu notre attention : la génération de cartes de hotspots⁴ en estimant la densité par la méthode des noyaux (« Kernel Density Estimation » ou KDE). Cette technique, très populaire en cartographie criminelle⁵ (Chainey, 2013), offre une visualisation spatialement continue de délits ponctuels, tout en exploitant le format raster⁶ bien connu du SIG. Or, aucun outil SOLAP n'est actuellement capable de gérer cette modélisation continue de l'espace.

Suite à ce constat, un second objectif de recherche a été défini dans le domaine du SOLAP au sens large. Puisque les outils SOLAP actuels ne sont capables de représenter l'espace que de manière discrète à travers le format vectoriel, l'élaboration d'un SOLAP raster spatialement continu constituerait un défi théorique dont l'enjeu dépasse le domaine de la cartographie criminelle. En effet, de nombreux autres domaines scientifiques impliquent une vision continue de l'espace sur de grands jeux de données raster : climatologie, écologie, télédétection, *etc.*

Cette recherche multidisciplinaire est donc finalement motivée par deux objectifs interconnectés. Le premier consiste à enrichir la vision spatiale des modèles SOLAP afin d'étendre leur domaine d'application, notamment en exploitant le modèle raster pour représenter l'espace de manière continue. Le second est l'élaboration d'un prototype SOLAP adapté aux besoins des analystes criminels, en particulier pour la génération de cartes de hotspots par KDE.

³ Les analystes stratégiques travaillent sur de grands jeux de données afin de comprendre et prévenir la criminalité de manière globale alors que les analystes opérationnels apportent un soutien à des enquêtes spécifiques. En parallèle, une autre thèse a d'ailleurs été réalisée à l'Unité de Géomatique sur le profilage géographique au niveau opérationnel (Trotta, 2014).

⁴ Aire géographique présentant une forte densité d'un phénomène.

⁵ Cette discipline, plus connue par sa traduction anglaise « crime mapping », exploite les techniques de l'analyse spatiale et de l'analyse criminelle au sens large (opérationnelle et stratégique).

⁶ Format de données spatiales où les pixels d'une image géoréférencée sont associés à une valeur dépendant du phénomène modélisé (par exemple : image satellitaire). Le format raster se distingue du format vectoriel où chaque entité spatiale est définie par une ou plusieurs primitives ponctuelles, linéaires ou polygonales.

Plan de la thèse

Ce travail est divisé en trois parties. La première dresse un état de l'art de notre question de recherche SOLAP afin d'élaborer notre hypothèse de travail. Le chapitre 1 retrace les bases théoriques de l'informatique décisionnelle, avec une attention particulière portée aux deux aspects couverts par notre recherche : la structure multidimensionnelle des entrepôts de données et l'OLAP. La description des différentes architectures « Relational OLAP » (ROLAP), « Multidimensional OLAP » (MOLAP) et « Hybrid OLAP » (HOLAP) vues à la fin du chapitre est un point essentiel sur lequel se basera notre hypothèse de recherche. Le chapitre 2 traite de la modélisation spatiale des données dans un SIG. Deux dualités des données spatiales y seront décrites. La première relève du phénomène modélisé qui peut être spatialement discret ou continu. La seconde différencie la modélisation vectorielle de la modélisation raster. Bien que la très grande majorité des SOLAP sur le marché soit vectorielle, nous relèverons que les données raster, réputées efficaces pour modéliser le continu, présentent tout de même de nombreux points communs avec les cubes de données MOLAP. Dans la suite du travail, nous nous focaliserons donc davantage sur le raster. Le chapitre 3 sera naturellement consacré au SOLAP, qui est un mélange des concepts OLAP et SIG vus aux deux chapitres précédents. Nous verrons qu'un outil SOLAP peut être conçu à partir d'un SIG (approche SIG dominant) ou d'un OLAP (approche OLAP dominant). Dans le chapitre 4, nous formulerons notre hypothèse de recherche. Celle-ci s'appuiera notamment sur une analyse de la littérature spécifiquement liée à la problématique de l'intégration de la continuité spatiale et du format raster dans le SOLAP.

La deuxième partie de la thèse est purement théorique et concerne principalement notre premier objectif de recherche : l'enrichissement de la modélisation spatiale du SOLAP. Dans le chapitre 5, nous présenterons un modèle SOLAP original à trois dimensions basé sur le format raster : le modèle raster multidimensionnel. Le formalisme employé sera fondé à la fois sur des concepts propres aux domaines du raster (pixel, index, *etc.*) et du SOLAP (fait, membre, *etc.*). La notion de cube raster y sera introduite, ainsi que la notion de raster dimensionnel permettant d'intégrer des entités spatiales à la volée dans l'analyse de l'utilisateur. A la fin du chapitre, nous décrirons les opérations SOLAP possibles avec ce modèle et les illustrerons par des exemples sur des données criminelles. Le chapitre 6 est une extension du modèle SOLAP raster du chapitre précédent. En exploitant le modèle relationnel (Codd, 1970 ; Kimball, 1996), nous serons en mesure de manipuler des hypercubes raster comprenant un nombre indéfini de dimensions (type de délit, jour de la semaine, heure de la journée, *etc.*). Puisque le modèle deviendra ainsi un ROLAP (OLAP relationnel), il sera alors possible de l'implémenter au moyen des SGBD raster relationnels vus au chapitre 2. Grâce au format raster, notre modèle SOLAP est capable de représenter des données spatialement continues. C'est ce que nous démontrerons au chapitre 7. Ce dernier chapitre se focalisera davantage sur la problématique des cartes de hotspots par KDE dans le domaine de la cartographie criminelle (second objectif de recherche).

Finalement, la troisième partie est consacrée à la validation de notre modèle SOLAP raster (chapitre 8). Un prototype opérationnel, basé sur une architecture web complètement « open source », y sera brièvement décrit. Des données de criminalité issues des villes de Londres et Seattle y sont intégrées, ainsi que des données de température de la surface lunaire (phénomène spatialement continu). Ce prototype est accessible via

l'adresse web suivante (un utilisateur peut être créé facilement en cliquant sur le lien « create user » en haut à gauche de la page d'accueil) :

<http://nolap01.ulg.ac.be/rastercube>

Le chapitre 8 fait office de tutoriel pour le prototype. A la fin du chapitre, une étude comparative nous permettra de mieux cibler la place du format raster dans la modélisation SOLAP de manière générale. Plusieurs perspectives de recherche seront ensuite évoquées avant de passer à la conclusion générale de la thèse.

PARTIE 1 : ETAT DE L'ART

« La connaissance s'acquiert par l'expérience, tout le reste n'est que de l'information. »

Albert Einstein

Chapitre 1 : informatique décisionnelle

L'informatique décisionnelle est la traduction du terme anglais « Business Intelligence » (BI). Cette discipline, apparue à la fin des années 80, fait référence à une catégorie d'outils permettant la collecte, la consolidation et l'analyse de données afin d'aider les entreprises dans le processus de prise de décision (Chee *et al*, 2009). Elle répond à un besoin de retirer rapidement de l'information hors de très grands jeux de données provenant de sources diverses. Actuellement, de nombreuses solutions BI propriétaires (SAP, 2015; Oracle, 2015; MicroStrategy, 2015) et « open source » (Pentaho, 2014; SpagoBI, 2014) sont proposées à des entreprises dans un but commercial, mais elles sont également utilisées dans des études scientifiques couvrant des domaines très variés : agriculture, foresterie, sismologie, environnement, criminologie, *etc.* Typiquement, une architecture BI comporte trois catégories d'outils (Figure 1) :

- Les ETL (« Extract », « Transform », « Load ») : ils permettent l'extraction et l'intégration de données provenant de sources généralement externes (bases de données, fichiers, *etc.*) dans un entrepôt de données.
- Les entrepôts de données (ED): les données y sont stockées selon une structure multidimensionnelle adaptée à l'analyse décisionnelle.
- Les outils d'analyse des données : OLAP (« OnLine Analytical Processing ») pour l'analyse manuelle, « data mining » pour l'analyse automatique et les outils de « reporting » pour la génération de rapports.

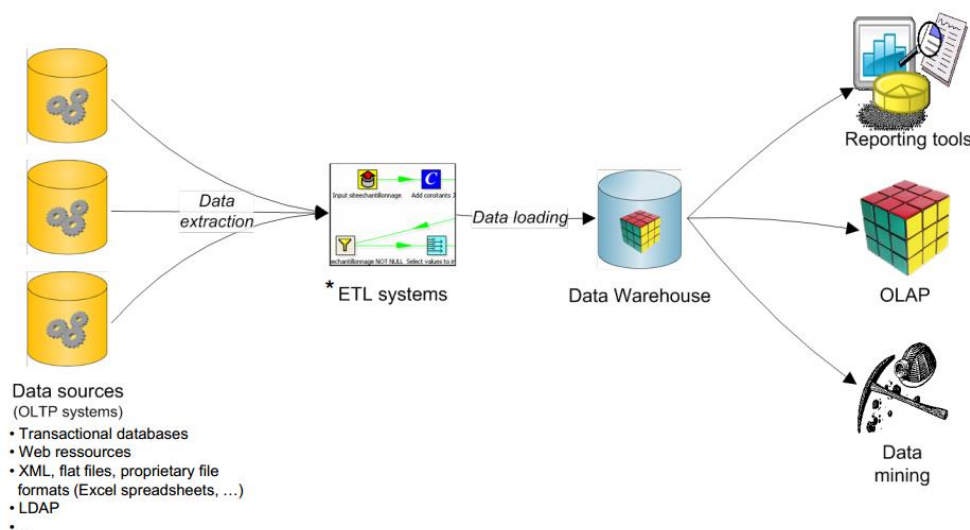


Figure 1. Architecture décisionnelle (Badard *et al*, 2009)

Ce chapitre est consacré à la description des différents éléments de l'informatique décisionnelle sur lesquels repose cette thèse : les ETL (section 1.1), les entrepôts de données (section 1.2), et les serveurs OLAP (1.3).

1.1. ETL

Les ETL autorisent un grand nombre d'opérations de manipulation de données. Comme l'indique l'acronyme ETL, celles-ci peuvent être regroupées en trois catégories :

- Les opérations d'extraction (« extract »)
- Les opérations de transformation (« transform »)
- Les opérations d'alimentation (« load »)

L'extraction et l'alimentation autorisent respectivement l'exportation et l'importation de données depuis/vers des fichiers (Excel, XML, shapefiles, *etc.*) ou des bases de données (gérées par Oracle, MySQL, PostgreSQL, *etc.*). Les opérations de transformation modifient la structure des données ou les données elles-mêmes afin de les adapter au fichier ou à la base de données qu'elles alimentent. Citons par exemple les jointures entre plusieurs sources, les calculs de nouveaux champs, les décompositions de champs, les modifications de type de champ, *etc.* Certains ETL comme GeoKettle (Spatialytics, 2014) proposent également des opérations propres aux données spatiales telles que jointures spatiales, définition de systèmes de coordonnées, transformations de coordonnées, *etc.*

Des scripts sont alors programmés par l'utilisateur via une interface graphique pour traiter de grands jeux de données automatiquement. Ceux-ci commencent toujours par l'extraction de données (« input ») et se terminent par l'alimentation (« output »). Entre les extractions et les alimentations, plusieurs transformations peuvent être programmées de manières séquentielle et/ou parallèle afin d'obtenir le résultat désiré. La Figure 2 montre un exemple de traitement sur des données spatiales avec l'outil GeoKettle. Un fichier de points (shapefile) est joint spatialement à une table de polygones extraite d'une base de données (chaque point est associé au polygone qui le contient). Ensuite, le résultat de la jointure spatiale est joint à un fichier Excel sur base d'un champ pour finalement alimenter une table d'une base de données.

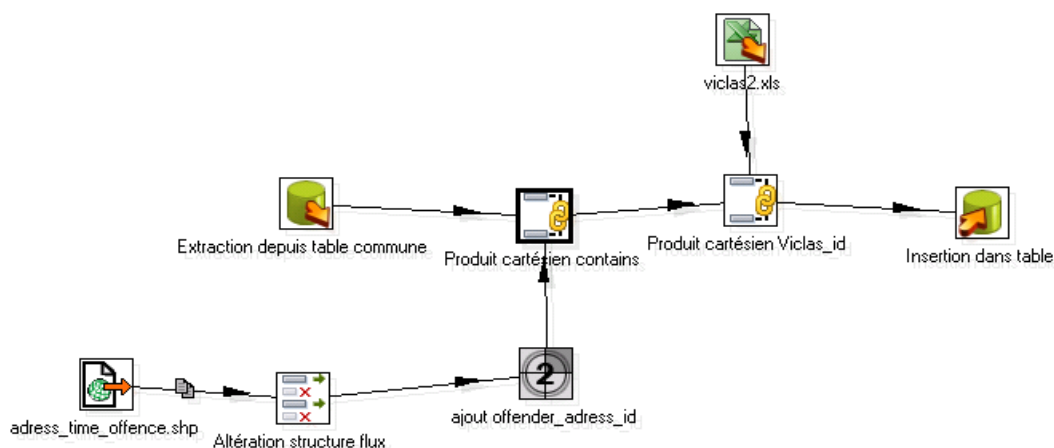


Figure 2. Exemple de traitement ETL avec l'outil GeoKettle (Spatialytics, 2014)

Les ETL sont donc des outils très polyvalents capables d'effectuer de nombreuses opérations à la chaîne sur des données diverses. Ils apportent des solutions à une problématique importante de l'informatique décisionnelle : l'intégration de grands jeux de données hétérogènes dans un ED.

1.2. Entrepôts de données

Les entrepôts de données sont le cœur de l'architecture décisionnelle. Les données détaillées y sont stockées selon une structure simple afin de faciliter leur exploitation dans le processus décisionnel. Il existe deux types d'ED : les ED relationnels et les ED multidimensionnels. Les ED relationnels étant les plus couramment utilisés, les prochaines descriptions de l'ED aux niveaux logique et physique concerneront les ED relationnels jusqu'à la section 1.3.5 dans laquelle nous aborderons le cas particulier des ED physiquement multidimensionnels.

Un ED relationnel peut être géré par n'importe quel système de gestion de base de données (SGBD) relationnel et peut donc être considéré comme une base de données présentant des caractéristiques propres à l'approche décisionnelle, appelée « OnLine Analytical Processing » (OLAP⁷). A l'opposé, l'approche transactionnelle, appelée « OnLine Transactional Processing » (OLTP), beaucoup plus courante dans la conception des bases de données, se distingue de l'OLAP selon plusieurs aspects (Proulx & Bédard, 2004).

Un OLTP est conçu pour supporter des activités quotidiennes sur les données. Le système est optimisé pour faciliter les mises à jour tout en maintenant l'intégrité des données. Les requêtes soumises sont simples et les utilisateurs sont nombreux. La normalisation des données est primordiale afin d'éviter les redondances et réduire la taille de la base de données au minimum. L'exemple typique de l'emploi d'un OLTP est l'encodage de produits par la caissière d'un supermarché.

Dans le cas de l'OLAP, les données sont stockées afin d'alimenter un historique dans lequel l'aspect temporel est primordial et où les mises à jour sont rares. Les utilisateurs sont moins nombreux puisqu'ils se situent au niveau stratégique⁸. Par conséquent, les requêtes sont plus sporadiques, mais elles sont aussi beaucoup plus complexes car elles impliquent des calculs d'agrégation sur de très grands volumes de données. Afin d'optimiser ces requêtes, la normalisation des données n'est plus systématique. Cela permet d'éviter de trop lourds calculs en stockant des données pré-agrégées à plusieurs niveaux de granularité. L'exemple typique d'un OLAP est l'opération réalisée par le décideur d'une grande chaîne de supermarchés lorsqu'il analyse l'évolution temporelle des ventes en fonction des types de produits et de la localisation des magasins.

Avec la venue de l'informatique décisionnelle, le concept de structure multidimensionnelle propre aux ED fait rapidement son apparition (Kimball, 1996). Largement appliquée dans l'informatique décisionnelle, elle est actuellement considérée

⁷ Le terme OLAP est à la fois utilisé pour caractériser l'approche décisionnelle des ED et le serveur permettant l'analyse manuelle des données de l'entrepôt (voir section 1.3).

⁸ Le plus haut niveau d'une organisation.

comme la structure de référence des entrepôts de données car elle présente d'importants avantages :

- Une certaine simplicité facilitant l'intégration de données hétérogènes et la compréhension de l'utilisateur lors de la construction de requêtes
- Une performance accrue des requêtes OLAP

Pour comprendre le fonctionnement d'une structure multidimensionnelle, et donc d'un entrepôt de données, il est essentiel de définir son vocabulaire. C'est précisément le but des prochaines sections. Après l'introduction des concepts de faits, mesures, dimensions, membres, hiérarchies et niveaux, nous pourrons aborder les différents types de modèles, conceptuels et logiques, utilisés dans la conception d'un ED. Nous clôturerons cette partie consacrée aux ED par la description de leurs instances : les cubes de données.

1.2.1. Dimensions, membres, hiérarchies et niveaux

Une dimension constitue un axe de l'analyse multidimensionnelle. Par exemple, une vente peut être caractérisée par un temps et par un type de produit afin de répondre à la question « Combien de raquettes de tennis ont été vendues au mois de janvier 2014? ». « Janvier 2014 » et « raquette de tennis » sont alors respectivement appelés membres des dimensions « temps » et « type de produit ». Une dimension est formée d'un ensemble de membres.

Une dimension peut aussi être caractérisée par une hiérarchie agrégative. Celle-ci est constituée de niveaux classant les membres selon une structure arborescente faite de relations parent-enfant. Dans l'exemple de la Figure 3, le schéma de la dimension temporelle (à gauche) contient 3 niveaux : « année », « mois » et « jour ». Des exemples d'instance de ce schéma hiérarchique sont décrits à droite de la Figure 3. Le membre « janvier 2014 », appartenant au niveau « mois », est le parent des membres du niveau « jour » faisant partie du mois de janvier : « 01-01-14 », « 02-01-14 », « 03-01-14 », *etc.* A l'inverse, « Janvier 2014 » est un enfant du membre « 2014 » du niveau « année ». En plus des liens relationnels définissant leur hiérarchie, les membres peuvent contenir différents attributs descriptifs pour enrichir l'analyse multidimensionnelle.

Une hiérarchie contient toujours un niveau particulier : le niveau « all ». Ce niveau n'est composé que d'un seul membre qui est parent de tous les autres membres de la dimension (racine de l'arbre). Ainsi, dans la Figure 3, « all » fait aussi bien référence à tous les jours, qu'à tous les mois, ou encore toutes les années.

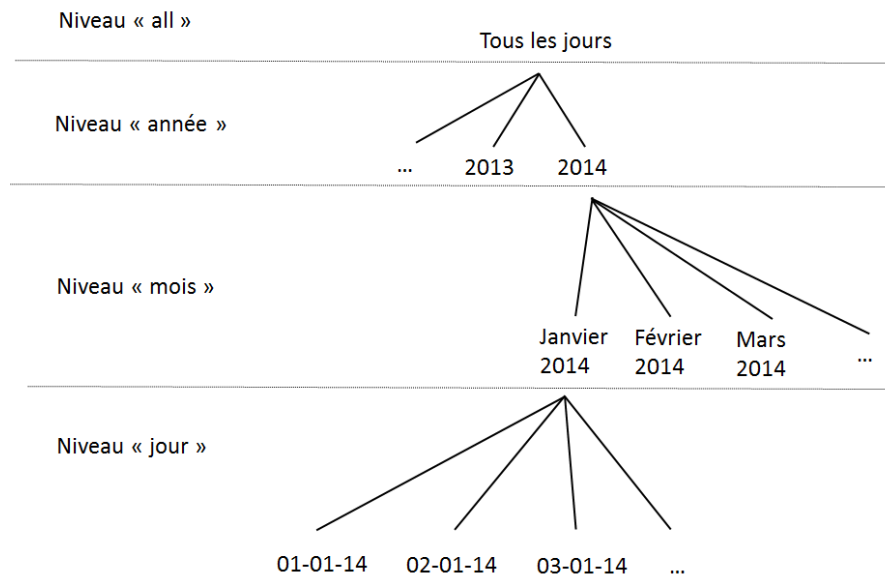


Figure 3. Exemple de dimension temporelle hiérarchisée

Plusieurs études ont montré que la hiérarchisation d'une dimension peut s'avérer assez complexe dans certains cas pratiques. Ainsi, une hiérarchie peut être stricte ou non-stricte, couvrante ou non-couvrante, symétrique ou asymétrique, unique ou multiple et statique ou dynamique⁹ (Pedersen *et al*, 2001 ; Malinowsky *et al*, 2006). Le Tableau 1 décrit et illustre ces dualités avec deux exemples extrêmes : une dimension temporelle hiérarchisée selon le calendrier grégorien « année-mois-jour » (cas le plus simple, en gris) et une dimension spatiale hiérarchisée selon le découpage administratif de la Belgique (cas le plus complexe, en blanc).

⁹ Liste non-exhaustive.

Caractéristique hiérarchique	Description	Exemple
Stricte	Un membre enfant a strictement un membre parent (cardinalités 1-N).	Tous les mois appartiennent à strictement une année.
Non-stricte	Au moins un membre a plusieurs parents (Cardinalités N-N).	La Région Flamande comprend la communauté flamande, la Région Wallonne comprend les communautés françaises et germanophones, et la Région de Bruxelles-Capitale est comprise à la fois dans la communauté française et la communauté flamande.
Symétrique	Seuls les membres du niveau le plus détaillé n'ont pas d'enfants.	Il n'existe pas de mois sans jours.
Asymétrique	Au moins un membre n'a pas d'enfant sans être au niveau le plus détaillé.	La Région Wallonne et la Région Flamande contiennent des provinces, mais pas la Région de Bruxelles-Capitale ¹⁰ .
Couvrante	Aucun membre ne saute de niveau.	Aucune année ne contient de jour sans mois.
Non-couvrante	Au moins un membre saute un ou plusieurs niveau(x).	La région de Bruxelles-Capitale n'a pas de province mais a des communes, alors que les autres régions ont des provinces qui ont elles-mêmes des communes.
Unique	Une seule hiérarchie décrit la dimension.	Une seule dimension temporelle : année-mois-jour.
Multiple	Plusieurs hiérarchies décrivent la dimension avec d'éventuels membres communs.	La Belgique est subdivisée en régions (Région Wallonne, Région Flamande et Région de Bruxelles-Capitale) et en communautés linguistiques (flamande, française et germanophone). Provinces et communes font à la fois partie des hiérarchies régions et communautés (niveaux communs).
Statique	Reste fixe par rapport aux autres dimensions.	Le calendrier est le même pour toutes les communes de Belgique (pas d'évolution en fonction de la dimension spatiale).
Dynamique	Évolue en fonction d'au moins une autre dimension.	La Belgique comptait 9 provinces avant 1995. Elle en compte maintenant 10 (évolution en fonction de la dimension temporelle).

Tableau 1. Différentes caractéristiques de hiérarchies dimensionnelles

¹⁰ Nous ne tenons pas compte des communes dans cet exemple, sinon la hiérarchie serait non-couvrante.

Finalement, notons que certaines dimensions particulières peuvent présenter des membres ordonnés. C'est le cas de la dimension temporelle où par exemple, le membre « 2014 » du niveau année se situe logiquement après le membre « 2013 ». Cet ordre permet d'afficher une évolution des données de manière logique pour l'utilisateur (par exemple, l'évolution annuelle des ventes de produit), mais s'avère également utile pour optimiser les requêtes soumises à un entrepôt physiquement multidimensionnel grâce à une indexation efficace, tout comme le regroupement des membres dans la hiérarchie (voir section 1.3.5.2).

1.2.2. Faits et mesures

Une mesure est une valeur, généralement numérique, associée à un membre d'une ou plusieurs dimensions. Par exemple, si le nombre de raquettes de tennis vendues au mois de janvier 2014 est 42, 42 est la valeur de la mesure « nombre de produits vendus » associée aux membres « raquette de tennis » et « janvier 2014 » des dimensions respectives « type de produit » et « temps ». Cette association de membres et de mesures est appelée « fait ». Le fait peut donc être vu comme une intersection des axes d'analyse d'un ED à laquelle correspondent une ou plusieurs mesure(s). En effet, la mesure « nombre de produits » peut être complétée par une deuxième mesure telle que « chiffre d'affaire ». Le fait de notre exemple peut alors être exprimé par « 42 raquettes de tennis ont été vendues en janvier 2014 pour un chiffre d'affaire de 420 € ».

Une mesure doit pouvoir être formulée de manière quantitative. En effet, la plupart des requêtes soumises à un ED font appel à des milliers, voire des millions de faits. Les résultats de ces requêtes sont alors agrégés dans des tableaux, des graphiques ou des cartes afin de livrer une information synthétique. Dans le cas de mesures non numériques, la fonction d'agrégation typiquement employée est le comptage comme le nombre de produits de notre exemple. Les opérations d'agrégation les plus courantes sur les mesures numériques sont la somme, la moyenne, le minimum et le maximum. Le choix d'une fonction d'agrégation peut être limité par la signification sémantique de la mesure. En effet, la somme des valeurs de températures n'a pas de sens alors que la moyenne, le maximum et le minimum en ont.

Les mesures les plus exploitables sont les mesures additives car elles peuvent être agrégées selon toutes les dimensions de l'ED (Kimball & Ross, 2013). Certaines mesures sont dites « semi-additives » lorsque leur agrégation a du sens seulement selon certaines dimensions. Par exemple, des soldes de comptes bancaires peuvent être sommés par clients, mais pas dans le temps. Comme cela pourrait arriver dans certaines structures d'ED (dimensions à hiérarchies multiples ou non-strictes, par exemple), une mesure ne peut pas être prise en compte plusieurs fois dans une agrégation.

Certaines mesures absentes de l'ED peuvent être calculées à la volée lors de la requête de l'utilisateur. On les appelle « mesures dérivées » (El-Gamal, 2007). Elles sont calculées à partir d'autres mesures ou à partir des propriétés de membres de dimensions. Par exemple, au lieu de sommer un prix unitaires pour calculer un chiffre d'affaire, on préférera le multiplier par le nombre de produits vendus : « chiffre d'affaire » = « nombre de produits » * « prix unitaire ». A l'opposé, les mesures stockées dans l'ED sont appelées « mesures atomiques ».

Dans la suite de cette thèse, nous ferons régulièrement référence à l'ensemble des faits les plus détaillés. Cet ensemble¹¹ regroupe les faits combinant un membre de chaque dimension de l'ED au niveau le plus fin (par exemple, le niveau « jour » de la Figure 3). Cette notion est importante car c'est à partir de cet ensemble minimum qu'un entrepôt est capable de calculer toutes les requêtes soumises par l'utilisateur, les mesures des faits moins détaillés pouvant logiquement être le résultat d'agrégation de mesures des faits aux niveaux inférieurs. Ainsi, un ED doit toujours stocker au minimum¹² l'ensemble des faits les plus détaillés¹³ (qui comprennent une ou plusieurs mesures atomiques). Dans la structure logique d'un ED relationnel typique, la table stockant l'ensemble des faits les plus détaillés est simplement appelée « table des faits ».

1.2.3. Modélisation conceptuelle

La modélisation conceptuelle est une étape cruciale dans la création d'une base de données (BD). Elle traduit une vision globale de la structure de la BD tout en restant indépendante des outils utilisés pour son implémentation. De plus, un modèle conceptuel (ou schéma) est généralement présenté sous forme graphique relativement simple, et constitue donc un excellent moyen de communication entre les informaticiens et les différents acteurs d'un projet.

Il existe des standards très puissants pour la conceptualisation des bases de données transactionnelles (OLTP). Les deux plus connus sont les modèles « Entité-Association » (EA) et le diagramme de classes d'« Unified Modeling Language » (UML). En revanche, la modélisation des entrepôts de données (OLAP) est nettement moins standardisée. Les modèles des OLTP ne sont pas directement adaptés aux OLAP en raison des concepts de la structure multidimensionnelle décrits précédemment (mesures, dimensions, faits, *etc.*). Certains auteurs proposent des extensions d'E-A (Sapia *et al*, 1999 ; Tryfona *et al*, 1999) alors que d'autres proposent des extensions d'UML (Trujillo *et al*, 2001 ; El Gamal *et al*, 2005 ; Zaamoune *et al*, 2013). Il existe aussi des formalismes complètement indépendants des modèles propres aux OLTP (Cabbibo & Torlone, 1998; Golfarelli *et al*, 1998; Tsois *et al*, 2001). Cependant, aucun ne permet la modélisation de toutes les particularités rencontrées dans un ED multidimensionnel : mesures semi-additives, mesures dérivées, dimensions non-strictes, dimensions non-couvrantes, changements de la structure de l'ED pour certains membres temporels, *etc.* El-Gamal (2007) propose une comparaison de ces différents modèles sur base de la présence/absence des concepts multidimensionnels évoqués précédemment.

A un haut niveau d'abstraction, la représentation du modèle multidimensionnel communément acceptée est le schéma en étoile (Kimball & Ross, 2013). L'ensemble des faits stockés dans l'entrepôt constitue l'élément central auquel sont reliées les différentes dimensions.

¹¹ Mathématiquement, il s'agit du produit cartésien des niveaux détaillés de chaque dimension.

¹² Nous verrons dans la section 1.2.5 que des faits moins détaillés peuvent être stockés pour des raisons d'optimisation de requêtes.

¹³ Dans certains travaux, les mesures sont également calculées par interpolation à partir de faits moins détaillés (Ahmed & Miquel, 2005; Bimonte, 2007; Vaisman & Zimányi, 2009 ; Zaamoune *et al* 2013).

1.2.4. Modélisation logique

Au niveau logique et relationnel, le schéma conceptuel en étoile présente quatre grandes variantes : le schéma en étoile logique, le schéma en flocon, le schéma en constellation et le schéma en galaxie. Dans tous les cas, un « tuple » de la table des faits correspond à un fait au niveau le plus détaillé et un tuple d'une table de dimension correspond à un membre.

Le simple schéma en étoile est le schéma logique le plus utilisé (Figure 4). Une table des faits centrale est reliée à des tables de dimensions dénormalisées : les différents niveaux hiérarchiques de dimensions sont regroupés dans la même table et sont considérés comme des attributs des membres. Cela implique une certaine redondance de l'information. En effet, dans ce type de modèle, les différents attributs reliés à un membre de dimension (par exemple, le nom de la catégorie d'un produit) sont répétés autant de fois qu'il y a d'enfants liés au membre en question (pour autant que la dimension soit hiérarchisée). Alors que cette caractéristique devrait être considérée comme une erreur de modélisation dans un OLTP, elle présente l'avantage de réduire le nombre de jointures et ainsi d'optimiser les requêtes soumises à un OLAP.

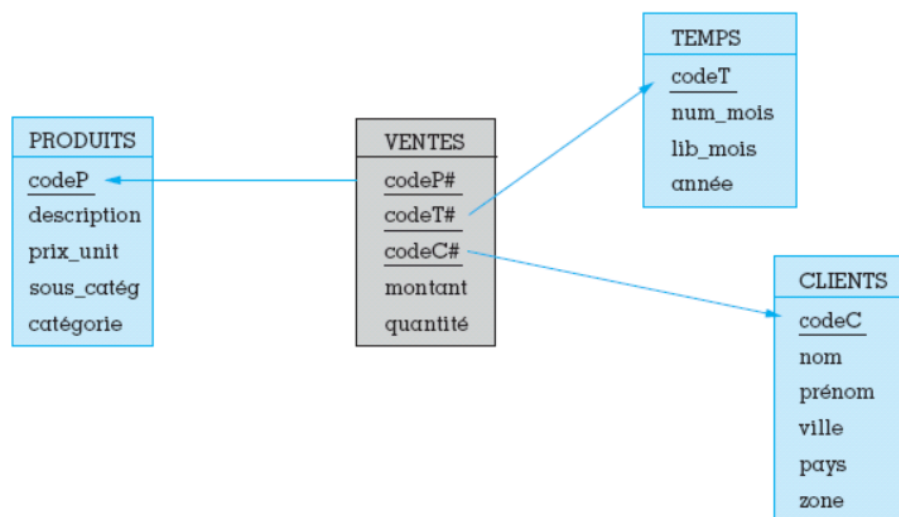


Figure 4. Schéma en étoile (Chrisment et al, 2005)

Inversement, une approche héritée des OLTP est retrouvée dans le schéma en flocon. Il s'agit d'une variante du schéma en étoile où les niveaux sont rejetés dans des tables normalisées reliées entre elles en fonction de leur hiérarchie dimensionnelle (voir Figure 5). L'espace de stockage est donc optimisé, mais les requêtes nécessitent beaucoup plus de jointures que dans le schéma en étoile. Un avantage important de ce schéma est la possibilité de modéliser des hiérarchies non-strictes (relations N-N entre niveaux de dimension) et multiples. Une hiérarchie multiple est d'ailleurs modélisée dans la Figure 5 où le membre « ville » possède deux parents : « pays » et « zones ».

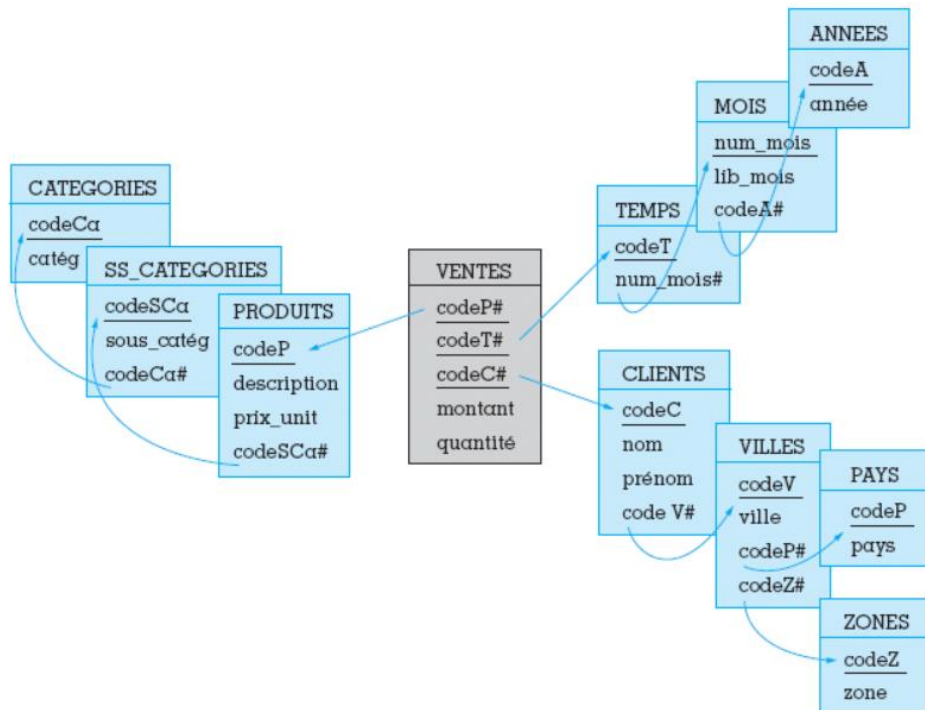


Figure 5. Schéma en flocon (Chrisment et al, 2005)

Les schémas en constellation et en galaxie sont des variantes des schémas en étoile ou en flocon dans lesquelles plusieurs tables des faits partagent des dimensions communes. Le schéma en constellation se différencie du schéma en galaxie par des tables des faits reliées entre elles, et offrant ainsi la possibilité de hiérarchiser le fait au même titre que la dimension (Miquel *et al*, 2002). Ces deux types de schémas peuvent être utilisés notamment pour matérialiser des cuboïdes de données (voir section suivante).

Enfin, les quatre schémas que nous venons d'évoquer peuvent éventuellement contenir des dimensions dégénérées. Il s'agit de dimensions considérées comme attributs directement dans la table des faits. Kimball & Ross (2013) déconseillent leur utilisation car elles alourdissent la table des faits qui occupe déjà la majorité de l'espace de stockage de l'entrepôt. Cela peut donc se traduire par la réplication d'attributs sur des millions d'enregistrements. Par conséquent, la dégénération ne devrait être utilisée que sur des dimensions non-hiérarchisées présentant une clé primaire comme unique attribut, ou sur des entrepôts présentant relativement peu de faits (quelques milliers tout au plus).

1.2.5. Cubes et cuboïdes de données

Un cube de données est une instance d'un schéma multidimensionnel d'ED. L'analogie avec un cube provient des trois dimensions retrouvées dans ce dernier pour traduire l'aspect multidimensionnel de l'ED. Lorsque le nombre de dimensions du cube est supérieur à trois, on parle parfois d'hypercube de données. Cependant, le terme cube englobe fréquemment à la fois les concepts d'hypercube et de cubes de données dans la littérature.

Un cuboïde¹⁴ (Han *et al*, 1998) est une vue matérialisée et agrégée d'un cube de données. En effet, dans l'utilisation d'un serveur OLAP (voir section 1.3), il est assez fréquent de répliquer des versions du cube agrégées selon certains niveaux de dimension afin d'optimiser les requêtes. Il devient alors plus simple pour le système de calculer des requêtes sur des niveaux de dimensions peu détaillés en agrégeant les membres du cuboïde adéquat, plutôt que d'agréger les faits aux niveaux les plus détaillés.

Théoriquement, un cube de données peut être décomposé en un treillis de cuboïdes hiérarchisé (Bimonte, 2007). Cette hiérarchie est organisée en fonction des niveaux de la structure multidimensionnelle du cube de base. Plus on monte dans la hiérarchie, plus les données sont agrégées, et plus les cuboïdes sont petits (réduction des enregistrements de la table des faits). Le cuboïde au sommet de la hiérarchie est appelé cuboïde apex et ne contient qu'un seul fait où l'ensemble des données du cube est agrégé. La Figure 6 montre un exemple de treillis de cuboïdes pour un ED présentant deux dimensions. A et 1 sont les niveaux les plus détaillés des dimensions respectives, alors que B et 2 sont les niveaux les moins détaillés. Les niveaux de dimensions agrégés sont indiqués sur les cuboïdes.

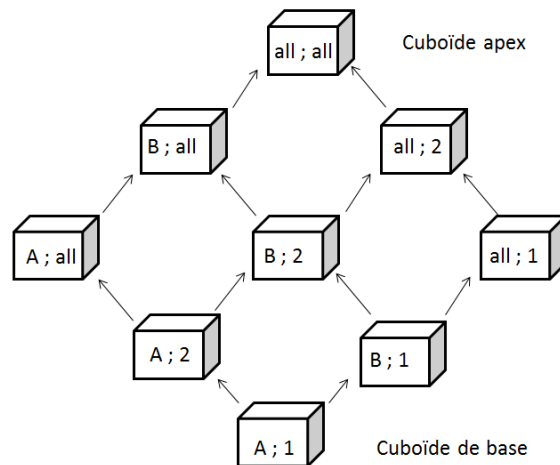


Figure 6. Treillis de cuboïdes

En pratique, il est pratiquement impossible d'entièrement décomposer un cube de données en un treillis de cuboïdes. En effet, si n est le nombre de dimensions d'un cube et que chaque dimension présente L_i niveaux (niveau « all » exclus), le nombre de cuboïdes nécessaires pour décomposer complètement le cube de données est

$$\prod_{i=1}^n (L_i + 1) \text{ (Bimonte, 2007)}$$

Le nombre de cuboïdes augmente exponentiellement avec le nombre de niveaux de dimensions différents et atteint donc rapidement de trop grandes valeurs pour la plupart des applications OLAP. Il devient alors nécessaire de trouver un compromis entre temps de

¹⁴ Dans le serveur OLAP Mondrian (Pentaho, 2014), un cuboïde peut être implémenté sous forme de table pré-agrégée.

réponse des requêtes, espace de stockage et maintenance du cube en matérialisant seulement quelques cuboïdes (Harinarayan *et al* 1996; Lin & Cuo, 2004).

Une autre difficulté rencontrée dans la construction de cuboïdes est le type de fonction d'agrégation associé aux mesures. Il en existe trois (Wang *et al*, 2007; Vaisman & Zimányi, 2009) :

- agrégation distributive : somme, comptage, min, max
- agrégation algébrique : moyenne
- agrégation holistique : médiane, mode, étendue

Des cuboïdes peuvent être construits en utilisant des fonctions distributives et algébriques, mais pas holistiques. En effet, d'après l'exemple de la Figure 6, en utilisant une fonction d'agrégation distributive ou algébrique, le résultat du cuboïde (A ; All) est le même en agrégeant aussi bien le cube de base (A ; 1) que le cuboïde (A ; 2). En utilisant une fonction holistique, les agrégations respectives du cube de base et du cuboïde (A ; 2) donneraient des résultats différents pour le cuboïde (A ; All).

1.3. OLAP

Alors que nous avons déjà mentionné plusieurs fois l'OLAP pour caractériser un entrepôt de données décisionnel par rapport à une base de données transactionnelle (OLTP), cet acronyme est également employé pour décrire le serveur permettant à la fois la définition de cubes de données à partir de l'ED, mais aussi le calcul des requêtes multidimensionnelles envoyées à l'ED par l'utilisateur. Alors que la section consacrée aux entrepôts de données portait essentiellement sur les aspects conceptuel et logique de la structure multidimensionnelle, le serveur OLAP se situe au niveau physique et utilisateur.

Après une brève description historique des caractéristiques de l'OLAP, nous nous attarderons sur les types d'interfaces et d'opérations qu'un OLAP offre à l'utilisateur à travers une programmation SQL ou MDX. La fin de cette section sera consacrée aux trois grandes approches de l'architecture OLAP : ROLAP, MOLAP et HOLAP.

1.3.1. Historique et caractéristiques

Le terme OLAP (« On-Line Analytical Processing ») est introduit par Codd *et al* (1993) dans une étude fondée en grande partie sur le modèle de tableur multidimensionnel d'EssBase (Arbor System). Initialement, l'OLAP était donc destiné à une structure de stockage physiquement multidimensionnelle (MOLAP¹⁵). Ce n'est que trois ans plus tard (Kimball, 1996) que le concept est étendu aux technologies de stockage relationnelles (ROLAP).

Codd définit plusieurs règles de conception pour l'OLAP, notamment les suivantes :

¹⁵ Voir section 1.3.5.2.

- Structure multidimensionnelle des données
- Transparence de l'emplacement physique de l'OLAP pour l'utilisateur
- Accessibilité de toutes les données par l'utilisateur pour son analyse
- Stabilité du serveur qui ne souffre pas du nombre dimensions dans l'analyse
- Architecture client-serveur
- Gestion complète des données clairessemées
- Gestion de plusieurs utilisateurs
- Flexibilité de l'édition des rapports
- Dimensionnement : les opérations appliquées sur une dimension sont théoriquement applicables sur les autres
- Analyse sans limites : le nombre de dimensions et de niveaux d'agrégation doit permettre à l'utilisateur des analyses très poussées.

L'aspect multidimensionnel étant une caractéristique clé du serveur OLAP, l'acronyme OLAP, alors jugé ambigu, est complété par FASMI : « Fast Analysis of Shared Multidimensional Information ». Celui-ci est introduit par Pendse (2007) sur le site web de l'OLAP report¹⁶. Cet acronyme alternatif est depuis lors largement accepté par la communauté BI comme définition de l'OLAP. Le mot « Fast » en particulier indique que les requêtes soumises à un OLAP ne devraient pas dépasser 5 secondes pour les requêtes simples et 30-60 secondes pour les requêtes les plus complexes.

1.3.2. Opérations OLAP

La navigation dans un cube de données a lieu au moyen de plusieurs opérations directement liées à la structure multidimensionnelle des données. Elles sont traduites par le serveur OLAP pour interroger le cube et livrer la réponse à l'utilisateur. Ces opérations peuvent être divisées en trois catégories. En voici quelques exemples :

- Les opérations de forage :
 - o « Roll-up » : agrège les mesures au niveau supérieur d'une dimension
 - o « Drill-down » désagrège les mesures au niveau inférieur d'une dimension
 - o « Drill through » : affiche les données détaillées d'un fait (mesures, dimensions, et attributs de dimensions)
- Les opérations de coupe :
 - o « Dice » : crée un « sous-cube » de données en supprimant certains membres de certaines dimensions
 - o « Slice » : fixe les agrégations sur un ou plusieurs membre(s) d'une dimension
- Les autres opérations :
 - o « Pivot » : échange les axes d'analyse dimensionnels de l'utilisateur

¹⁶ L'OLAP report, maintenant appelé « BI Verdict », est une communauté mondiale d'utilisateurs de produits BI complètement indépendante de toute société commerciale. Son but est l'évaluation des produits BI pour guider les utilisateurs dans le choix de l'outil adapté à leurs besoins.

- o « Drill across » : compare des données provenant de plusieurs cubes partageant le même schéma et la même instance¹⁷¹⁸

Notons qu'une opération « roll up » sur le membre « all » (niveau le moins détaillé) d'une dimension équivaut à ne pas tenir compte de la dimension dans l'analyse. En effet, en l'absence de cuboïde pour simplifier les agrégations, lorsqu'une dimension du cube est omise dans l'analyse, les données sont agrégées sur l'ensemble des membres du niveau le plus détaillé de la dimension. D'ailleurs, si ce type de requête est exprimée en SQL, inclure tous les membres d'un niveau de dimension dans la clause « WHERE » équivaut à n'inclure aucun membre de cette dimension.

Comme nous le verrons plus tard, il existe d'autres opérations s'appliquant à des cas spécifiques, notamment en ce qui concerne l'information spatiale.

1.3.3. Interfaces OLAP

Un serveur OLAP étant un produit ciblant les utilisateurs du niveau stratégique, leur exploitation doit être simple et abordable par des non-informaticiens. L'interface OLAP constitue donc un aspect important de l'architecture décisionnelle. Les prochaines sections sont consacrées aux deux¹⁹ grands types d'interfaces OLAP : la table de pivot et le graphique. Le serveur OLAP doit être capable d'alimenter ces deux interfaces et de gérer leurs interactions l'une avec l'autre : une modification de la table de pivot entraîne une modification du graphique, et *vice versa*.

1.3.3.1. Table de pivot

L'interface utilisateur la plus courante pour la navigation dans un cube de données est la table de pivot. Il s'agit d'un tableau multidimensionnel interactif dans lequel l'utilisateur a la possibilité d'afficher des mesures agrégées à travers des opérateurs OLAP tels que ceux vus à la section 1.3.2. Sa manipulation de type « point & click » est simple et intuitive car elle ne nécessite aucune connaissance en programmation (SQL ou autre). De plus, la vue qu'elle offre sur les données traduit fidèlement la structure multidimensionnelle de l'ED. L'utilisateur peut afficher plusieurs dimensions et mesures à la fois, organisées en lignes et en colonnes, en fonction de ses préférences.

La Figure 7 montre un exemple de forage de type « drill down » sur une mesure « nombre de délits » agrégée sur les membres d'une dimension spatiale représentant les entités administratives de la Belgique. Le « drill down » est appliqué sur la province d'Anvers pour afficher les données désagrégées selon ses membres enfants défini dans la hiérarchie de la dimension : les communes appartenant à la province d'Anvers. Notons que des opérations OLAP sont applicables sur des dimensions n'apparaissant pas dans la table

¹⁷ En réalité, les conditions du « drill across » ont été étendues dans (Abello *et al*, 2002) et (Gomez *et al*, 2012).

¹⁸ Dans certains travaux, l'opération « drill across », ou « forage latéral », est définie comme un changement de membre sur un niveau de dimension. Par exemple : le passage de l'agrégation des données de 2013 à 2014.

¹⁹ Il existe une troisième interface importante qui est propre à l'OLAP spatial (SOLAP) : la carte. Celle-ci sera abordée à la section 3.2.

de pivot. Par exemple, un slice sur la dimension « type de délit » permettrait d'agréger uniquement les délits appartenant à la catégorie « viols ».

offence place	Mesures
ANTWERPEN	29
BRABANT-WALLON	11
HAINAUT	24
LI+GE	13
LIMBURG	11
LUXEMBOURG	3
NAMUR	8
OOST-VLAANDEREN	12
VLAAMS BRABANT	36
WEST-VLAANDEREN	5

drill down

offence place	Mesures
ANTWERPEN	29
ANTWERPEN	11
BUGGENHOUT	1
LIER	3
MALLE	2
MOL	3
MORTSEL	1
OLEN	1
RANST	2
SCHILDE	4
ZANDHOVEN	1
ZOERSEL	
BRABANT-WALLON	11
HAINAUT	24
LI+GE	13
LIMBURG	11
LUXEMBOURG	3
NAMUR	8
OOST-VLAANDEREN	12
VLAAMS BRABANT	36
WEST-VLAANDEREN	5

Figure 7. Opération « drill down » dans une table de pivot

1.3.3.2. Graphiques

Bien que très complète, la vue des données offerte par la table de pivot est purement numérique et offre une faible vision synoptique de l'information. C'est pourquoi elle est souvent complétée par des graphiques. La Figure 8 montre un graphique obtenu à partir de la vision des données issue de la Figure 7.

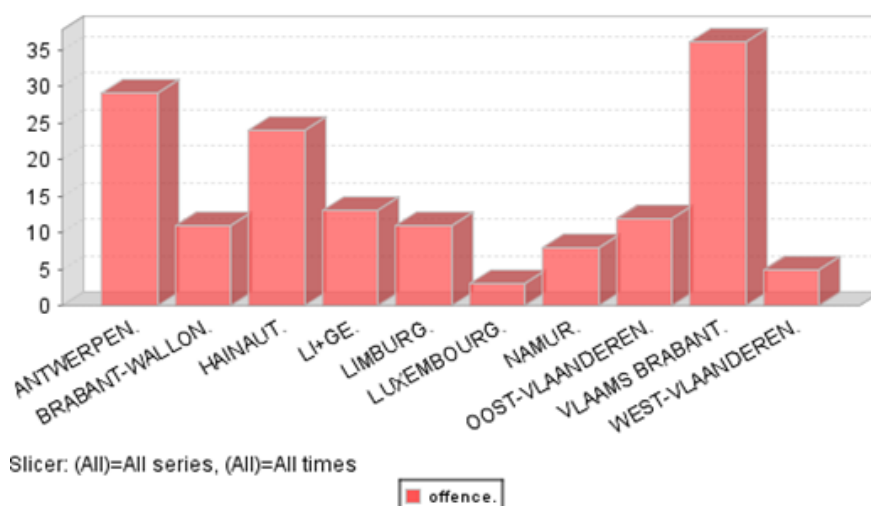


Figure 8. Graphique issu d'un cube de données : représentation des délits par province

1.3.4. Le langage MDX

Bien que la plupart des requêtes OLAP puisse être programmée en SQL, langage de référence des bases de données relationnelles, leur formulation s'avère relativement complexe. En particulier, les requêtes nécessitant des agrégations dans un tableau multidimensionnel (telle qu'une table de pivot) sont exprimées de manière maladroite. Un langage dédié à l'OLAP est alors présenté par Microsoft en 1996 : le langage MDX (« Multidimensional Expressions »).

Dès 1997, MDX est utilisé par Microsoft dans son extension OLAP du SGBD SQL Server (Microsoft, 2014a). Par la suite, d'autres outils OLAP intègrent le langage, y compris dans le domaine « open source » (Pentaho, 2014). Ainsi, MDX est aujourd'hui considéré comme le langage de référence de l'OLAP, en particulier pour les architectures MOLAP (voir section 1.3.5.2).

La syntaxe MDX, finalement assez proche du SQL, se base sur une structure de type « SELECT ... FROM ... WHERE ». Le « SELECT » contient les mesures et les membres de dimensions à afficher en rangées et en colonnes (équivalent au « GROUP BY » en SQL), le « FROM » contient le cube de données définissant la structure multidimensionnelle des données et le WHERE permet d'appliquer des coupes sur plusieurs membres de dimensions. Voici un exemple de requête calculant les résultats des Figure 7 et Figure 8:

```
SELECT {[Measures].[offence]} ON COLUMNS,  
       {[offence place].[province].members} ON ROWS  
FROM [offence]
```

Au-delà de ce simple exemple, MDX permet le calcul de requêtes assez complexes, notamment par la construction de mesures dérivées, la sélection de membres par rapport à leur parent ou enfant hiérarchique, *etc.*

1.3.5. Architectures OLAP

Il existe trois grandes catégories de serveur OLAP : « Relational OLAP » (ROLAP), « Multidimensional OLAP » (MOLAP) et « Hybrid OLAP » (HOLAP). ROLAP se base sur un ED relationnel avec une structure similaire à celles vues à la section 1.2.4 alors que MOLAP se base sur un ED multidimensionnel. Dans un HOLAP, les deux technologies ROLAP et MOLAP sont présentes afin d'en tirer les avantages respectifs. Les prochaines sections sont consacrées à la description de ces architectures avec une comparaison de leurs avantages et inconvénients.

1.3.5.1. ROLAP

ROLAP (Kimball, 1996) est l'architecture OLAP la plus utilisée (Figure 9). Les données sont stockées dans un ED relationnel selon un des schémas logiques mentionnés à la section 1.2.4. Entre l'interface utilisateur et l'ED se situe le moteur OLAP présentant plusieurs fonctions:

- La définition des cubes de données selon une structure multidimensionnelle
- La traduction de requêtes multidimensionnelles en langage relationnel (SQL)
- La communication avec l'interface OLAP

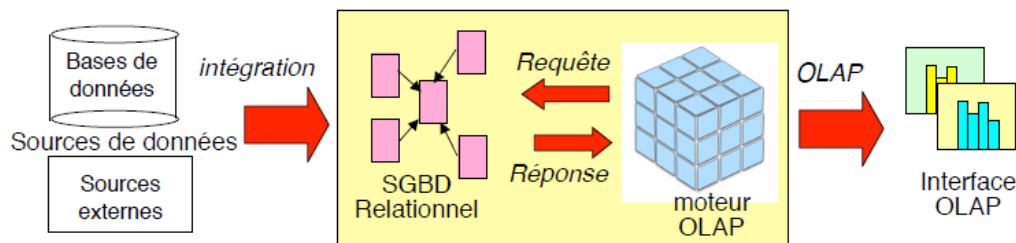


Figure 9. Architecture ROLAP (Espinasse, 2010)

Les requêtes sont optimisées par des techniques d'indexation telles que les listes inversées, les vecteurs de bits ou les indexations de jointure (Espinasse, 2010), et par le stockage de cuboïdes (voir section 1.2.5). Dans un contexte ROLAP, les cuboïdes sont aussi appelés « tables de pré-agrégation » (Pentaho, 2014).

L'utilisation de la technologie relationnelle est l'atout principal de cette architecture. Ainsi, elle bénéficie des avantages d'un modèle de données de référence qui a largement démontré son efficacité depuis une quarantaine d'années (Codd, 1970). Il existe de nombreux systèmes de gestion de base de données relationnels (SGBDR) permettant de stocker de très grands volumes de données simples, mais aussi plus complexes telles que les données spatiales. La simplicité de la structure logique de l'ED (étoile, flocon, etc.) minimise le nombre de jointures lors des requêtes de l'utilisateur.

En revanche, le temps de réponse du système est plus lent que dans un MOLAP puisque la structure physique des données par tables et enregistrements n'est pas directement assimilable à celle d'un cube de données (d'où la présence du moteur OLAP). De plus, l'expression des requêtes multidimensionnelles en SQL peut s'avérer complexe comme nous l'avons vu précédemment.

Mondrian (Pentaho, 2014) est un bon exemple de serveur ROLAP dans le milieu « open source ». L'utilisateur y connecte un entrepôt de données relationnel structuré selon un des schémas multidimensionnels décrits à la section 1.2.4, et y importe un fichier XML de définition de son cube de données. Dans ce fichier, les concepts de la structure multidimensionnelle (faits, mesures, dimensions, membres, hiérarchies, etc.) sont définis par rapport aux tables et attributs de l'entrepôt de données. De cette manière, le serveur Mondrian est capable de traduire les requêtes MDX de l'utilisateur en SQL pour fournir le résultat des requêtes OLAP.

1.3.5.2. MOLAP

Dans une architecture MOLAP (Figure 10), la façon de stocker les données est très différente de celle du ROLAP. L'entrepôt de données n'est plus relationnel, mais multidimensionnel, et les données sont donc physiquement²⁰ multidimensionnelles.

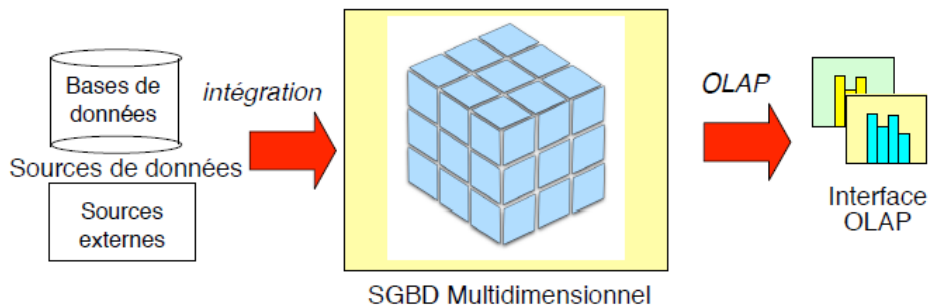


Figure 10. Architecture MOLAP (Espinasse, 2010)

Les membres de chaque dimension sont ordonnés et indexés. Par exemple, dans la dimension « type de produit », le membre « raquette de tennis » est 0, le membre « raquette de ping pong » est 1, etc. Ainsi, un cube de données à n dimensions peut être stocké dans un tableau à n dimensions. Chaque agrégat au niveau le plus détaillé est associé à une cellule indexée par n membres de dimension et la valeur d'une cellule (mesure atomique) associée à son adresse (liste de membres) correspond à un fait. Les agrégations sont donc très rapides puisqu'elles se font en lignes et en colonnes (Figure 11).

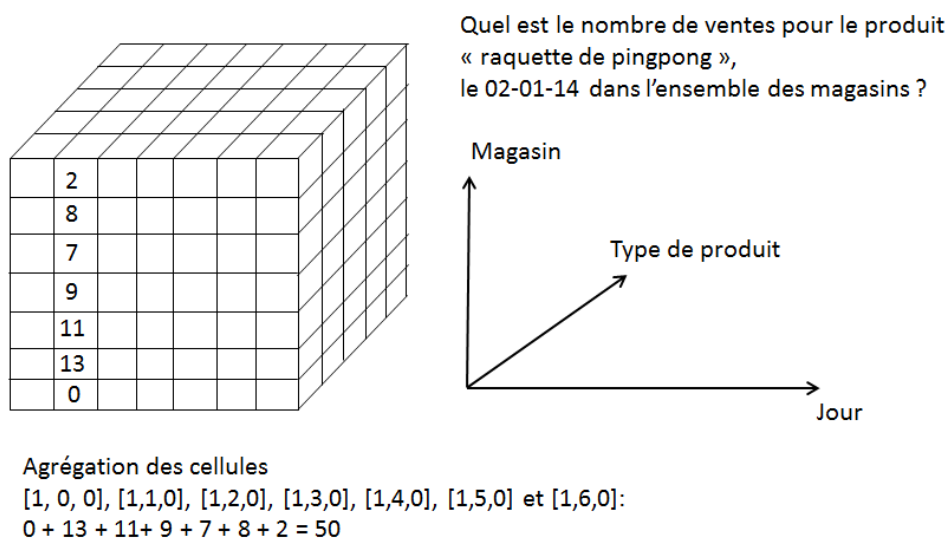


Figure 11. Agrégation dans un cube MOLAP

²⁰ Par « physiquement », nous voulons dire qu'il s'agit de la façon la plus évidente et la plus directe pour une machine de modéliser un cube de données.

Contrairement au ROLAP, ce type de stockage peut impliquer un nombre considérable de cellules vides. En effet, dans l'exemple de la Figure 11, si environ 10% des types de produits sont vendus chaque jour dans chaque magasin, le cube de données contiendrait 90% de cellules vides. Ce problème est connu sous le nom de « gestion de la faible densité » et peut être résolu par des techniques de compression et d'index spécifiques. Hélas, ces techniques ne fonctionnent que pour des cubes de données de maximum trois dimensions.

Les requêtes peuvent être optimisées par le calcul de cuboïdes, mais aussi par partitionnement des dimensions en sous-cubes de même taille. Par exemple, un cube d'un an peut être partitionné en quatre sous-cubes d'un trimestre. Les requêtes sont ainsi effectuées par agrégations partielles des cellules de plusieurs sous-cubes.

Le temps de réponse d'un MOLAP est plus rapide que celui d'un ROLAP en raison de l'accès aux cellules par leurs index. Cependant, l'espace de stockage est moins rentabilisé, et la performance des cubes décroît rapidement avec le nombre de dimensions. Le MOLAP est donc moins efficace que le ROLAP avec des données caractérisées par beaucoup de dimensions (cubes de faible densité). Les mises à jour et la redéfinition de cubes sont aussi plus problématiques. Puisque, contrairement au ROLAP avec le modèle relationnel, cette technologie ne s'appuie pas sur un modèle de données de référence, toutes les opérations de stockage et de manipulation de données doivent être redéfinies dans le système.

Actuellement, la plupart des solutions MOLAP font partie du domaine propriétaire. De ce fait, il existe presque autant de méthodes d'interrogation des cubes de données multidimensionnels que d'outils, même si la plupart sont basés sur des variantes de MDX. SQL Server (Microsoft, 2014a) est un des plus populaires. Ce dernier est également capable de supporter une architecture HOLAP (voir section suivante).

1.3.5.3. HOLAP

Dans une architecture HOLAP (Figure 12), les deux approches ROLAP et MOLAP sont présentes afin d'en tirer les avantages respectifs. De grands volumes de données détaillées et caractérisées par un grand nombre de dimensions sont stockés dans un ED relationnel. Un moteur ROLAP est présent pour directement interroger les cubes de données relationnels. En plus de cela, certains jeux de données sont pré-agrégés dans des cuboïdes MOLAP (« data marts » ou magasins de données) avec une granularité plus grossière et/ou une dimensionnalité réduite afin de garantir des temps de réponses courts sur certaines requêtes.

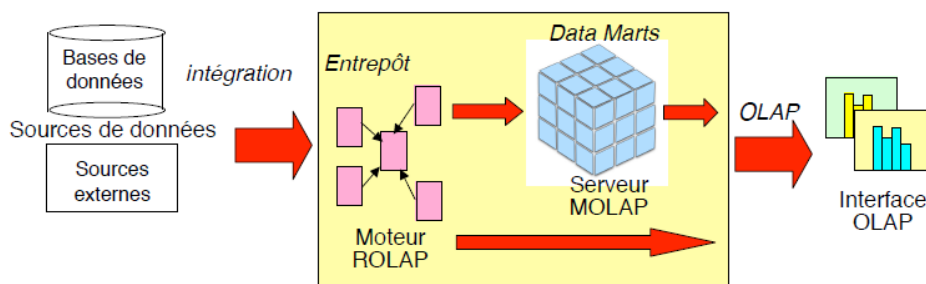


Figure 12. Architecture HOLAP (Espinasse, 2010)

1.3. Conclusion

Le but de ce chapitre était d'introduire l'OLAP en le replaçant dans son contexte : l'informatique décisionnelle. OLAP est donc un serveur permettant à un décideur d'analyser manuellement de grands jeux de données hétérogènes à travers des opérations d'agrégation. Pour que cette analyse soit possible, les données provenant de sources diverses sont intégrées via un outil ETL dans un entrepôt de données présentant une structure multidimensionnelle.

Afin de comprendre la structure multidimensionnelle, les termes clés de son vocabulaire ont été définis : dimension, membre, hiérarchie, niveau, fait et mesure. Après avoir présenté le schéma en étoile des modèles multidimensionnels au niveau conceptuel, nous avons abordé les différentes manières de traduire ce schéma au niveau logique pour un ED relationnel : schéma en étoile, en flocon, en constellation et en galaxie. La section sur les entrepôts de données et la structure multidimensionnelle a ainsi pu être clôturée par la définition de l'instance d'un schéma multidimensionnel : le cube de données. Celui-ci peut être décomposé en cuboïdes de données pré-agrégées. En effet, nous avons vu que l'approche OLAP (à ne pas confondre avec le serveur OLAP), contrairement à l'approche OLTP, autorisait la redondance de l'information et la dénormalisation des tables afin d'accélérer le temps de réponse du système, quitte à consommer plus d'espace.

Après avoir introduit les entrepôts de données et la structure multidimensionnelle, nous avons pu définir les caractéristiques du serveur OLAP : opérations de manipulation de cubes de données, types d'interfaces utilisateur (tables de pivot et graphiques), langage MDX de référence et les trois grands types d'architectures OLAP (MOLAP, ROLAP et HOLAP). Alors que le ROLAP utilise un entrepôt de données relationnel, le MOLAP exploite les données directement stockées sous forme de tableaux multidimensionnels par le serveur OLAP : on parle alors d'entrepôt de données (ou de cube de données) physiquement multidimensionnel. Le ROLAP permet le stockage de grands volumes de données détaillés, mais le MOLAP délivre des réponses plus rapidement. HOLAP est une architecture combinant à la fois les caractéristiques du ROLAP et du MOLAP afin d'en tirer les avantages respectifs.

Chapitre 2 : SIG et format raster

Pour aborder la notion de système d'information géographique (SIG), il convient d'abord de définir un système d'information au sens large. Un système d'information (SI) est « *une association de personnes, machines, données et procédures travaillant ensemble pour collecter, gérer et distribuer de l'information aux individus d'une organisation*²¹ » (Workboys & Duckham, 2004, p.1). Un SIG, spécialisé dans l'information à caractère spatial, est défini comme « *un SI informatisé permettant l'acquisition, la modélisation, le stockage, la récupération, le partage, la manipulation, l'analyse et la présentation de données géoréférencées* » (Workboys & Duckham, 2004, p. 2).

Les données d'un SIG sont stockées dans une base de données transactionnelle et relationnelle. Une table contient donc des enregistrements caractérisés par des attributs de type sémantique²² comme dans une base de données classique (par exemple : le numéro d'une parcelle). Souvent²³, la base de données sur laquelle repose le SIG est dite spatiale lorsqu'elle permet la définition de tables spatiales présentant un ou plusieurs attributs complexes : les attributs spatiaux. Ces attributs particuliers contiennent des informations sur la forme et la localisation des entités géographiques modélisés par la table spatiale (par exemple : l'emprise spatiale d'une parcelle). Grâce aux attributs spatiaux, les enregistrements d'une table présentent des propriétés particulières permettant leur visualisation cartographique, mais aussi leur traitement par des opérations relevant du domaine de l'analyse spatiale. Actuellement, la plupart des SGBD proposent une extension pour la gestion de l'information géographique : PostGIS, Oracle Spatial, MySQL, etc.

Un attribut spatial est défini au minimum selon les deux dimensions planimétriques dans un système de coordonnées de référence : coordonnées cartographiques (X, Y) pour les données projetées ou coordonnées géographiques (latitude, longitude) pour les données non-projetées. Dans le cas particulier d'un SIG 3D, l'attribut spatial est également défini selon la troisième dimension de l'espace (hauteur) dans un système de référence altimétrique (Laurini & Servigne, 2008). Si cette information est gérée par un attribut simple, le SIG est uniquement capable de représenter la hauteur par entité qu'il modélise. Dans ce cas, on parle parfois de SIG 2,5 D (coupe 2D dans un espace 3D).

Il ne faut pas confondre SIG et SIG-logiciel (Donnay, 2005b). Un SIG-logiciel est un outil fournissant des fonctions d'analyse spatiale et de représentation cartographique, parfois très complètes, en utilisant des données importées par l'utilisateur. En aucun cas, il ne présente à lui seul toutes les caractéristiques d'un système d'information : contrôle de la qualité des données, partage de l'information, modélisation adaptée aux besoins de l'organisation, etc. « *Un SIG ne s'achète pas, il se conçoit* » (Donnay, 2005b, p. 50). En revanche, les SIG-logiciels peuvent faire partie de l'architecture d'un SI et ils constituent

²¹ Le terme organisation comprend les entreprises et les gouvernements, mais aussi des groupes d'individus plus diffus tels que des communautés scientifiques partageant des intérêts communs.

²² Les informations sémantiques comprennent toutes les informations non-géométriques (non-spatiales).

²³ Ce n'est pas toujours le cas. En effet, ArcGIS (ESRI, 2012) offre la possibilité de spatialiser des tables provenant de SGBD non-spatiaux via son outil ArcSDE.

d'ailleurs de très bonnes interfaces client pour interroger les bases de données spatiales gérées par un SGBD. En effet, la plupart d'entre eux fournissent des connexions à des serveurs de manière directe pour les SGBD spatiaux les plus populaires, ou de manière indirecte via des protocoles d'échange de données sur le web (WMS²⁴, WFS²⁵, WCS²⁶). Le SIG-logiciel propriétaire le plus connu est ArcGIS (ESRI). Le domaine « open source » compte aussi quelques références du genre : Quantum GIS, Grass, OpenJump, *etc.*

Il existe deux approches de modélisation de l'information spatiale : l'approche vectorielle et l'approche raster. Les deux ont des avantages et inconvénients vis-à-vis des deux grands types de phénomènes spatiaux : discrets et continus. Après avoir présenté ces deux dualités (section 2.1), nous définirons la notion de couverture spatiale qui est un concept normalisé par l'ISO (2005) et l'OGC (Open GIS Consortium, 2006), commun au vecteur et au raster, et commun au discret et au continu (section 2.2). Nous nous focaliserons ensuite sur le format raster car celui-ci représente un aspect important de cette recherche. Ainsi, nous décrirons quelques caractéristiques essentielles du raster (algèbre de carte et méthodes d'optimisation des traitements, section 2.3), puis analyserons deux stratégies de modélisation physique du raster dans deux SGBD relationnels populaires (section 2.4) : Oracle GeoRaster (Oracle, 2003) et PostGIS raster (Racine & Cumming, 2011). Ce chapitre sera naturellement finalisé par une conclusion.

2.1. Dualités de l'information spatiale

2.1.1. Dualité des phénomènes géographiques : discrets et continus

Conceptuellement, les phénomènes géographiques sont classés en deux grandes catégories : les phénomènes spatialement discrets et les phénomènes spatialement continus (OGC, 2006). Les phénomènes discrets présentent des frontières et une extension spatiale relativement bien définie. Citons par exemple les bâtiments, les routes ou les états. A l'opposé, les phénomènes continus peuvent être mesurés ou décrits uniquement à une position particulière de l'espace (et éventuellement du temps). La valeur d'un phénomène continu en un point est directement liée aux valeurs se trouvant à proximité (autocorrélation spatiale), et cette valeur est toujours quantitative. Citons par exemple la température, le relief ou la concentration d'une molécule dans l'air.

Du point de vue de la modélisation physique, un phénomène spatialement continu est toujours stocké de manière discrète car il est impossible de stocker (et de mesurer) l'infinité de valeurs engendrée par le phénomène. On parle alors de champ (« field ») ou de « grid ». Il existe deux stratégies de modélisation d'un champ (Collet, 1992). Dans la première stratégie (Figure 13a), un certain nombre de points d'échantillonnage sont stockés et un point quelconque du champ est calculé par interpolation sur base des points d'échantillonnage (bilinéaire, bicubique, *etc.*). Dans la seconde stratégie (Figure 13b), l'espace est divisé en mailles associées à une valeur, et la valeur d'un point quelconque du champ est assimilée à la valeur de la maille dans laquelle il est inclus.

²⁴ « Web Map Service ».

²⁵ « Web Feature Service ».

²⁶ « Web Coverage Service ».

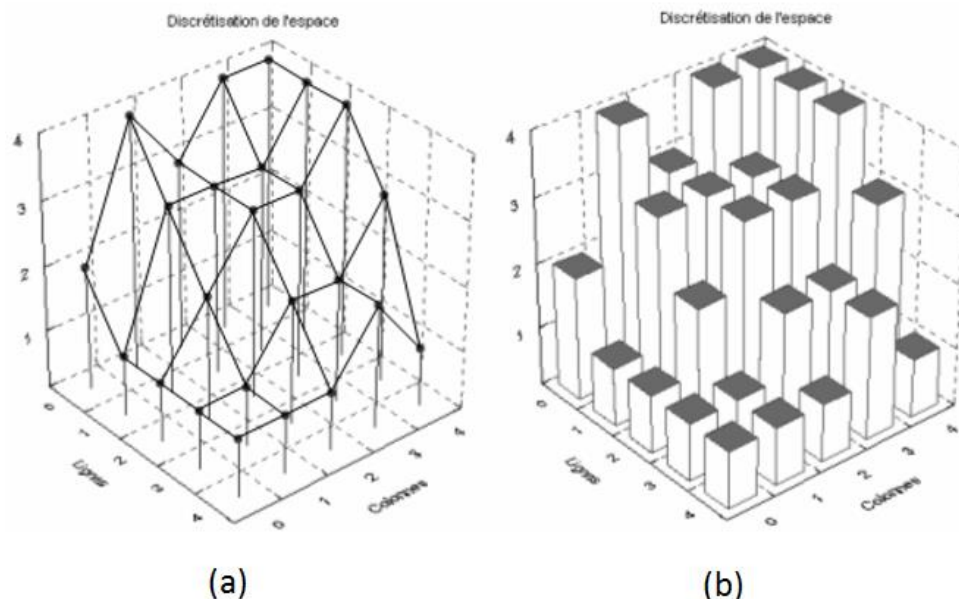


Figure 13. Deux stratégies de modélisation d'un champ (Donnay, 2005a). a) Interpolation sur base de points d'échantillonnage. b) Discrétisation de l'espace en mailles

Rappelons que le concept de champ, même lorsqu'il représente une hauteur, est très différent du SIG 3D. En un point de coordonnées (x, y) , un champ n'est capable de représenter qu'une seule valeur sur la troisième dimension h (par exemple, l'altitude au niveau du sol par rapport à un référentiel altimétrique), alors que le SIG 3D représente l'infinité²⁷ de valeurs de la dimension h (l'altitude au niveau du sol, mais également l'altitude de tous les points au-dessus et en dessous du sol).

2.1.2. Dualité de la modélisation spatiale : vecteur et raster

2.1.2.1. Données vecteur

L'approche vectorielle modélise la géométrie des entités géographiques à l'aide de trois primitives (Figure 14) : le point, la polyligne et le polygone. Un point est stocké par un couple de coordonnées. Une polyligne est stockée par un ensemble ordonné de points qui forme une chaîne de segments. Un polygone est constitué d'une ou plusieurs polyligne(s) fermée(s) et caractérisée(s) par un espace intérieur et extérieur. Notons qu'une entité spatiale est éventuellement modélisable par plusieurs primitives du même type (multipoints, multipolygones, multipolygones).

²⁷ Dans les limites du domaine spatial altimétrique.

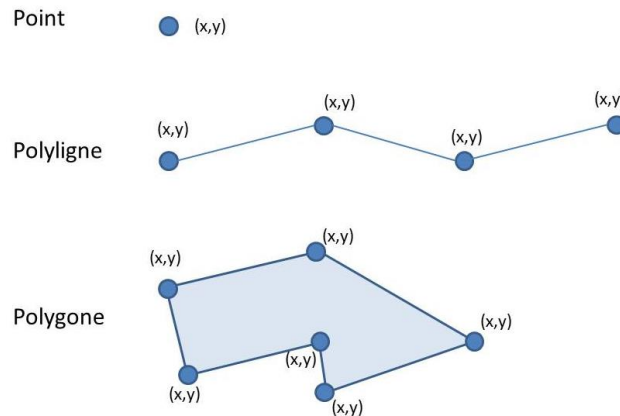


Figure 14. Les trois primitives vectorielles

Au-delà de la définition géométrique d'entités dans l'espace géographique, la topologie décrit les relations spatiales relatives des entités les unes par rapport aux autres. Un modèle topologique bien connu est le modèle des 9 intersections (Egenhofer, 1989). Celui-ci définit les 9 relations spatiales possibles entre deux primitives, appelées cellules²⁸, dans une matrice d'intersection. Les intersections représentent une valeur binaire entre l'intérieur (A^0), la frontière (∂A) et l'extérieur (A^-) des cellules A et B. La Figure 15 montre les 8 relations topologiques possibles entre deux polygones ainsi que leur description dans la matrice des 9 intersections.

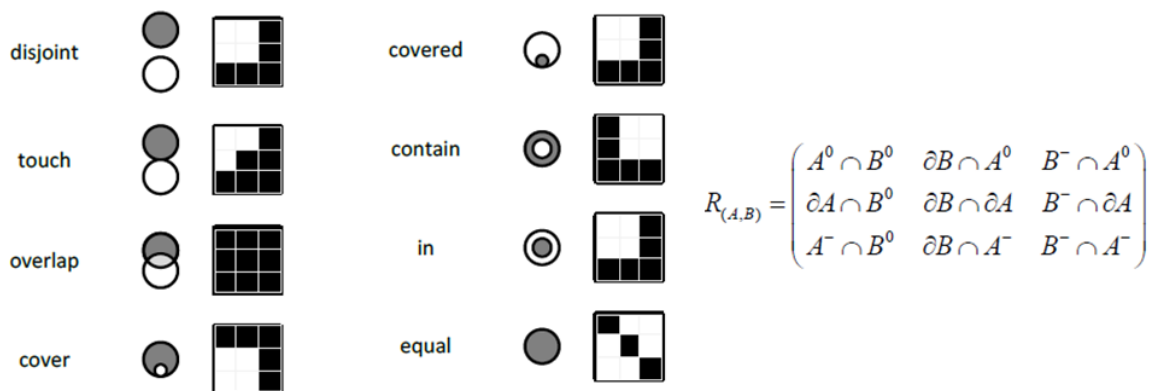


Figure 15. Les 8 relations topologiques entre deux régions et leur description d'après la matrice des 9 intersections (Hallot, 2012)

La topologie est importante car elle est utilisée dans de nombreuses opérations d'analyse spatiale. Des imprécisions géométriques sur les données vectorielles entraînent une incohérence topologique affectant l'exactitude des résultats de ces opérations. C'est pourquoi le contrôle de la qualité géométrique des données est une tâche très importante d'un SIG vectoriel, même si cette tâche peut s'avérer assez lourde puisqu'elle n'est pas toujours automatisable.

²⁸ A ne pas confondre avec la cellule (pixel) d'un raster. La cellule d'Egenhofer peut prendre la forme de n'importe quelle primitive du mode vecteur.

Notons que bien que la topologie soit facilement calculable à la volée, elle peut aussi être stockée dans la base de données. Cela permet de résoudre les opérations topologiques de manière purement relationnelle (sans faire appel aux propriétés des attributs spatiaux) et d'éviter la redondance de l'information géométrique. En effet, si deux polygones ont une frontière commune, cette frontière ne devrait être définie qu'une seule fois. En mode vecteur, la topologie peut être définie par une structure faite de nœuds et d'arcs. Un arc est une arête orientée et délimitée par un nœud de début (« from node ») et un nœud de fin (« to node »), et un nœud peut faire partie de plusieurs arcs. Les polygones et polygones sont alors définis par les arcs qu'ils contiennent. La notion d'intérieur et d'extérieur d'un polygone est définie par l'identifiant des polygones situés respectivement à gauche et à droite de chaque arc. Notons que la topologie peut également être définie sur la dimension temporelle (Hallot, 2012).

Dans une base de données spatiale, le modèle relationnel peut être étendu en y intégrant la notion de « géométrie » pour la définition d'un type d'attribut. Un attribut géométrique est associé à une primitive vectorielle, et définit ainsi la spatialité de l'ensemble des tuples d'une relation. Lorsqu'un enregistrement est associé à un attribut géométrique, il devient possible de lui appliquer des traitements exploitant aussi bien sa spatialité que ses attributs sémantiques. Dans la majorité des cas, l'ensemble des enregistrements d'une table spatiale constitue une couche couvrant un domaine sémantique particulier. Par exemple, les villes sont stockées dans une table de points, les cours d'eau dans une table de polygones, les entités administratives dans une table de polygones, *etc.*

2.1.2.2. Données raster

L'approche raster modélise l'information spatiale à l'aide d'un tableau (Fiume, 1989; Collet, 1992; Morain & Lopez, 1996 ; Demers, 2001; Workboys & Duckham, 2004) tridimensionnel qui ressemble très fort à la structure physique d'un cube de données MOLAP (voir section 1.3.5.2). Un raster contient un ensemble de cellules (ou pixels) qui constituent l'unique primitive permettant de décrire un phénomène géographique. Par conséquent, plusieurs types d'entités spatiales peuvent être représentés dans un seul raster (ponctuelles, linéaires ou zonales). Une cellule est une valeur numérique indexée par les trois dimensions du raster: la rangée, la colonne et la bande²⁹ (coordonnées cellules du raster). Dans l'espace raster (voir plus bas), la cellule correspond à un carré et l'ensemble des cellules couvre le domaine spatial³⁰ du raster. La Figure 16 montre un exemple de raster à 3 rangées, 3 colonnes et 3 bandes (27 cellules au total).

²⁹ En général, le terme « bande » est utilisé pour décrire une fenêtre du spectre électromagnétique provenant d'une image satellitaire. Dans le contexte de la modélisation raster au sens large, nous utilisons ce terme pour identifier la troisième dimension d'un raster.

³⁰ Infinité de points comprise dans un espace planimétrique délimité par un rectangle, ou « Minimum Bounding Rectangle » (MBR).

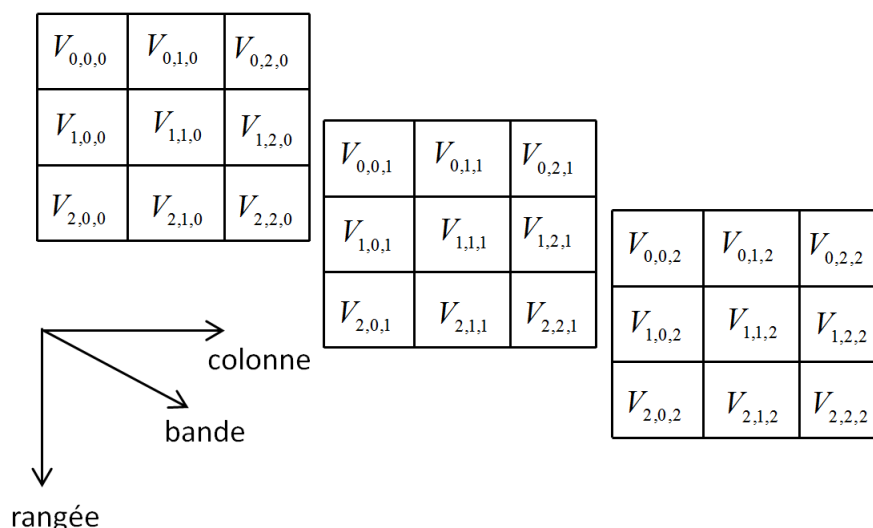


Figure 16. Exemple de raster de dimensions 3*3*3

Les coordonnées « cellule » sont des entiers ordonnés d'après les trois dimensions du raster. Les dimensions « rangée » et « colonne » correspondent à la position d'une cellule dans un espace planimétrique local³¹, appelé espace raster (Oracle, 2003). Il s'agit donc de dimensions spatiales. La dimension « bande », quant à elle, peut être associée à n'importe quelle dimension sémantique, ou éventuellement à une troisième dimension spatiale : la hauteur par rapport à un système de référence altimétrique.

Afin de relier les cellules à des entités géographiques, le raster doit être associé à des paramètres de transformation du système de coordonnées de l'espace raster (colonne, rangée) vers le système de coordonnées de référence de l'espace géographique associé (X, Y). Dans la grande majorité des cas³², ce changement d'espace peut être résolu par une transformation affine simplifiée (Donnay, 2010) à 4 paramètres: translation sur la colonne, translation sur la rangée, homothétie sur la colonne et homothétie sur la rangée³³. Les paramètres d'homothétie sur la colonne et la rangée correspondent respectivement à l'inverse de la résolution spatiale du raster en X et en Y dans un système de coordonnées de référence. Autrement dit, la résolution spatiale est le rapport du nombre de pixels par unité de longueur du système de coordonnées de référence en X et en Y. Une haute résolution implique donc des pixels de « petite taille » et une faible résolution des pixels de « grande taille ». Pour un même domaine spatial, la réduction de la résolution entraîne logiquement une diminution du nombre total de cellules, et donc de la taille physique du raster. Comme nous l'expliquerons à la section 2.3.2.2, cette importante propriété est exploitée dans les structures raster pyramidales pour optimiser les traitements à différentes échelles d'analyse.

³¹ Dans l'espace raster, l'origine du système est dans le coin supérieur gauche du tableau. L'axe Y pointe donc vers le bas.

³² Nous ne tenons pas compte des cas rares où les axes « rangée » et « colonne » de l'espace raster ne sont pas alignés respectivement avec les axes y et x de l'espace géographique (homothétie complète à 6 paramètres).

³³ Vu que l'axe rangée du raster pointe vers le « bas » (Sud) et que l'axe Y des système de coordonnées de référence pointe vers le « haut » (Nord), l'homothétie est toujours négative sur la colonne et positive sur la rangée.

Les valeurs des cellules sont toujours numériques, mais elles peuvent couvrir plusieurs domaines de valeurs (quantitatif ou qualitatif). Elles sont directement liées au phénomène spatial modélisé par le raster. Il peut par exemple s'agir d'une intensité de rayonnement électromagnétique dans le cas d'une image satellitaire, d'une valeur de température ou d'un entier renvoyant à une légende des types de sols³⁴.

Alors que les différentes relations topologiques en mode vecteur sont relativement nombreuses, doivent être contrôlées, et nécessitent des opérations géométriques pour être calculées à la volée, la topologie raster, beaucoup plus simple (topologie du jeu d'échec), est stockée implicitement. En effet, grâce à la structure tabulaire du raster, les cellules présentant une proximité d'après leurs coordonnées dans l'espace raster (rangée et colonnes) présentent la même proximité relative dans l'espace géographique (Morain & Lopez Baros, 1996). Cette importante propriété du raster est exploitée dans de nombreux algorithmes d'analyse spatiale, notamment l'algèbre de carte (voir section 2.3.1). Si la taille d'une cellule d'un raster obtenu par rasterisation³⁵ d'une couche vectorielle, ou par tout traitement « vecteur vers raster » (exemple : calcul de la densité par la méthode des noyaux), est supérieure à la précision géométrique vectorielle, les éventuels problèmes topologiques issus du mode vectoriel n'ont pas de conséquence en mode raster.

En revanche, le raster présente un inconvénient important par rapport au vecteur : la gestion de l'absence de phénomène. En effet, les régions de l'espace considérées comme vides d'un point de vue sémantique ne sont tout simplement pas définies en mode vecteur. En raster, ces régions couvertes par le raster sont gérées par une valeur « no data ». Néanmoins, dans le cas d'un raster qualitatif, la multiplication des valeurs « no data » contigües peut être gérée par des algorithmes de compression.

Dans une base de données relationnelle, l'intégration du raster est possible grâce à la définition d'un type d'attribut raster présentant diverses propriétés. La modélisation logique et physique d'un raster dans une base de données fera l'objet d'une analyse plus approfondie dans la section 2.4.

2.1.3. Bilan

Le but de cette section était de présenter deux grandes dualités de l'information spatiale. D'un côté les phénomènes spatialement discrets et continus, et de l'autre côté les modélisations vecteur et raster. Nous avons vu que le vecteur et le raster offrent des avantages l'un par rapport à l'autre au niveau de la modélisation spatiale. Le vecteur présente une certaine rentabilité au niveau sémantique puisqu'il ne stocke que les régions spatiales où le phénomène modélisé est présent, contrairement au raster. De plus, la précision géométrique des entités est conceptuellement³⁶ absolue lorsqu'elles sont

³⁴ Oracle (2003) et Donnay (2005a) font d'ailleurs la distinction entre les images géographiques, provenant d'un capteur sensible à une partie du spectre électromagnétique, et les champs géographiques, résultant de l'interpolation de plusieurs points sondés pour traduire un phénomène spatialement continu. Les deux sont modélisables en raster.

³⁵ Processus de conversion des données vectorielles en données raster.

³⁶ En réalité, cette précision géométrique est limitée par le codage des valeurs des coordonnées, par le nombre de points utilisé pour modéliser une entité, et bien entendu par l'acquisition des données.

modélisées en vecteur alors qu'elle dépend de la résolution spatiale lorsqu'elles sont modélisées en raster. En revanche, le vecteur nécessite une gestion de la topologie pouvant s'avérer assez lourde, alors que cette topologie est incluse implicitement dans le raster. Le raster est également plus rentable au niveau géométrique puisqu'aucune coordonnée n'est réellement stockée³⁷, mis à part deux points nécessaires au calcul des quatre paramètres de géoréférenciation³⁸ (espace raster vers espace géographique).

Les deux types de phénomènes spatiaux, continus et discrets, peuvent être modélisés aussi bien par du vecteur que par du raster. Cela dit, la gestion de l'absence de phénomène par le vecteur nous amène naturellement à le préférer au raster pour les phénomènes discrets. De plus, en raster, les entités zonales et linéaires discrètes sont représentées par un certain nombre de pixels présentant la même valeur qualitative, ce qui entraîne une certaine redondance de l'information (gérable par des algorithmes de compression). Cependant, certains traitements spatiaux sont beaucoup plus efficaces sur des phénomènes discrets modélisés en raster, notamment l'analyse multi critères ou les algorithmes de propagation (Demers, 2001; Kasprzyk *et al*, 2012).

En ce qui concerne l'information spatialement continue, le raster s'avère être un excellent candidat. En effet, comme nous l'avons vu, un phénomène spatialement continu peut être modélisé en discrétisant l'espace en mailles de valeurs connues, ce qui est fait implicitement dans le raster (« grid raster »). Le grid raster est composé de mailles carrées³⁹ et régulières, et un point quelconque de l'espace appartient à une des mailles d'où il tire sa valeur. La localisation (absolue ou relative) des mailles dans l'espace géographique est retrouvée grâce à leur position dans le tableau raster et par conséquent, le stockage des coordonnées ou de la topologie des différentes mailles n'est pas nécessaire. Par contre, si le phénomène modélisé n'est présent que sur certaines régions de l'espace étudié (par exemple, une rivière), un certain nombre de mailles sont associées à la valeur « no data ». Cette problématique n'existe pas en vecteur puisque comme nous l'avons vu, l'absence d'information au niveau conceptuel se traduit par l'absence d'information au niveau physique.

Lorsqu'un champ continu est modélisé en vecteur, les deux stratégies (points d'échantillonnage et mailles) sont applicables. Dans un cas comme dans l'autre, la géométrie des points d'échantillonnage ou des mailles, et éventuellement la topologie, doivent être stockées⁴⁰. Le stockage de l'information géométrique est donc beaucoup plus important dans un champ vectoriel par rapport à un champ raster. En revanche, le mode vecteur permet une définition beaucoup plus souple du champ et autorise notamment un découpage irrégulier de l'espace tel qu'on peut le voir dans des polygones de Thiessen ou dans une triangulation de Delaunay.

³⁷ Les index rangée et colonne sont logiquement déduits par la position d'une cellule dans le tableau

³⁸ Uniquement dans les cas les plus simples et les plus courants (voir section 2.1.2.2).

³⁹ Carrée dans l'espace raster, mais pas nécessairement dans l'espace géographique. Par exemple : résolution spatiale différente en X et n Y.

⁴⁰ Notons qu'il est possible de modéliser un champ vectoriel régulier en stockant les points d'échantillonnage ou les mailles d'une manière tabulaire assez similaire au raster. Ce genre de structure très particulière associe les avantages du raster dans le mode vectoriel puisque les géométries n'ont plus besoin d'être stockées et que les mailles ne sont pas nécessairement carrées ou rectangulaires. Faut-il considérer cette structure comme du vecteur, comme du raster ou comme les deux? La question reste ouverte.

2.2. Couverture spatiale

Indépendamment du type de phénomène (discret et continu) et du type de modélisation spatiale (vecteur ou raster) dans un SIG, la norme ISO 19123 (ISO, 2005) et l'OGC (2006) définissent la notion de couverture spatiale (« coverage »). La couverture est une sous-classe de la métaclasse « feature » (ISO 19101) qui associe toutes les positions possibles d'un espace spatiotemporel borné (domaine spatiotemporel) à un vecteur de valeurs d'attributs (étendue de la couverture). Il s'agit donc à la fois d'un « feature » et d'une fonction (ISO, 2005).

Le domaine spatiotemporel de la couverture est associé à un système de coordonnées de référence et comprend l'infinité de points se trouvant à l'intérieur de l'enveloppe convexe qui délimite le domaine. Les valeurs attributaires de ces points sont donc retrouvées par interpolation (bilinéaire, au plus proche voisin, *etc.*) à partir des valeurs qui sont stockées dans les attributs du « feature ». Le « feature » est un découpage de l'espace basé sur des entités discrètes, ou sur une tessellation spatiale régulière (rectangulaire, hexagonal) ou irrégulière (triangulation de Delaunay, polygones de Thiessen, *etc.*). La Figure 17 montre les sous-classes de la classe « coverage » qui illustrent les possibles découpages spatiaux d'une couverture.

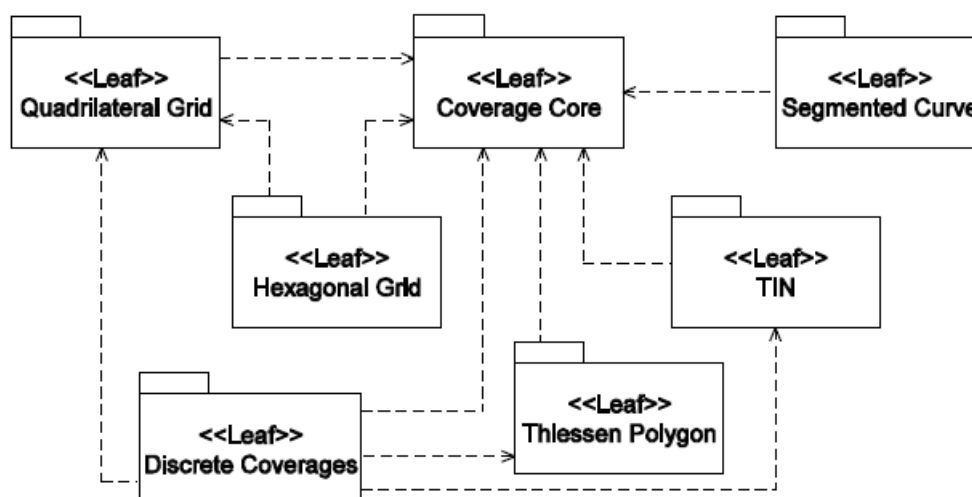


Figure 17. Les différents découpages spatiaux héritant de la classe « coverage » (OGC, 2006)

L'étendue de la couverture concerne le domaine de ses attributs. Ceux-ci peuvent être discrets ou continus, et une couverture peut associer plusieurs fonctions pour le calcul de divers attributs (par exemple pour associer une température et un taux d'humidité à toutes les positions du domaine spatiotemporel).

Une couverture peut décrire aussi bien des phénomènes spatialement discrets que des champs continus, et elle peut être modélisée en vecteur ou en raster, le raster étant particulièrement concerné par la classe « quadrilateral grid ».

2.3. Caractéristiques du format raster

2.3.1. Algèbre de carte

L'algèbre de carte, ou « map algebra » (Tomlin, 1983, 1991 ; Demers, 2001) est une collection de fonctions pour le traitement de couches d'une même couverture spatiale. Celles-ci permettent la modélisation de différents problèmes géographiques et d'extraire de nouvelles informations (« output ») hors des données initiales (« input »). Bien que l'algèbre de carte soit un concept indépendant de la modélisation des données spatiales (Gomez *et al*, 2012), celle-ci est le plus souvent appliquée à des données raster en raison de leur structure maillée particulièrement adaptée à ces types de traitements. Notons que puisque la traduction de l'anglais vers le français du terme « map » en « carte » peut sembler ambiguë dans ce contexte, nous préférons utiliser le terme « couche » pour les données manipulées dans l'algèbre de carte.

L'algèbre de carte est très largement utilisé en analyse spatiale, notamment en analyse multicritère (Eastman *et al*, 1995) et plus généralement dans l'aide à la décision, car elle est à la base de traitements assez variés qui apportent une réponse « cartographique » à une question spatiale. Elle se décompose en trois types de fonctions : les fonctions locales (Figure 18a), les fonctions focales (Figure 18b) et les fonctions zonales (Figure 18c).

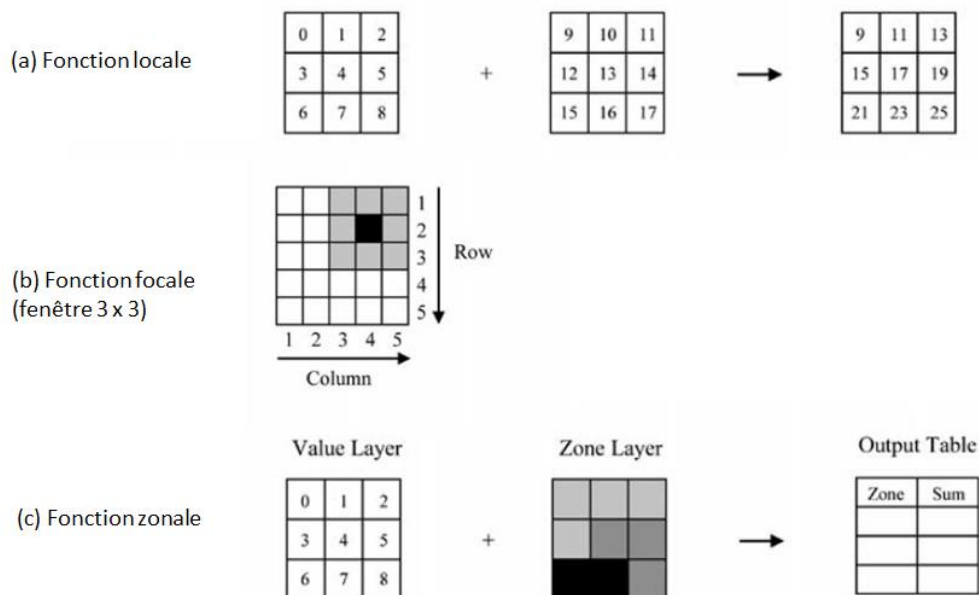


Figure 18. Les trois types de fonctions d'algèbre de carte (Mennis *et al*, 2005)

- Fonctions locales⁴¹ : la valeur attribuée à un endroit de la couche en sortie est le résultat d'un traitement appliqué sur la valeur de cet endroit sur une ou plusieurs couche(s) en entrée. Par exemple : production d'une couche de température moyenne de l'année à partir des couches de températures journalières moyennes.
- Fonctions focales : la valeur attribuée à un endroit de la couche en sortie est le résultat d'un traitement appliqué dans le voisinage de cet endroit dans une ou plusieurs couche(s) en entrée. Par exemple : production d'une couche de valeur de pentes à partir d'un modèle numérique de terrain (MNT).
- Fonctions zonales⁴² : la valeur attribuée à un endroit de la couche en sortie est le résultat d'un traitement appliqué à l'ensemble de l'entité géographique (« zone layer ») à laquelle appartient cet endroit, pour le phénomène étudié (« value layer »). Par exemple : production d'une couche des altitudes moyennes par commune à partir d'un MNT et d'une couche des entités communales.

Alors que l'algèbre de carte a longtemps été associée à la manipulation de couches de données spatiales bidimensionnelles, Mennis *et al* (2005) redéfinissent les trois fonctions de base dans un cube de données à trois dimensions : les deux dimensions planimétriques et le temps. Depuis, comme nous le verrons dans la section 4.2, l'algèbre de carte a été reprise dans plusieurs travaux traitant des entrepôts de données spatiales et du SOLAP (McHugh, 2008; Vaisman & Zimányi, 2009; Bimonte & Kang, 2010; Gomez *et al*, 2011; Gomez *et al*, 2012; Plante, 2014).

2.3.2. Optimisation des traitements raster

La taille physique d'un raster est un aspect important de ce modèle de données car elle influence considérablement la longueur des traitements qui lui sont appliqués (par exemple : une opération d'algèbre de carte). Cette taille dépend de deux paramètres fondamentaux :

- Le codage des valeurs des pixels : 1 bit, 8 bits, 16 bits, *etc.*
- Le nombre de pixels définis dans le raster

Le codage des valeurs est un problème classique de base de données auquel nous ne nous attarderons pas. En revanche, la problématique du nombre de pixels peut être gérée par deux méthodes spécifiques que nous allons brièvement décrire dans les prochaines sections : les tuiles et les pyramides⁴³. Rappelons que le nombre total de pixels est le résultat de la multiplication du nombre de rangées, du nombre de colonnes et du nombre de bandes du raster (voir section 2.1.2.2). Le nombre de rangées et de colonnes est lui-même dépendant de l'emprise spatiale du raster (ou le domaine spatiale de la couverture) et de sa résolution spatiale.

⁴¹ Dans notre modélisation SOLAP raster (chapitre 5), les fonctions locales d'algèbre de carte sont à la base des agrégations non-spatiales.

⁴² Dans notre modélisation SOLAP raster (chapitre 5), les fonctions zonales d'algèbre de carte sont à la base des agrégations spatiales.

⁴³ Ces deux méthodes peuvent être combinées.

2.3.2.1. Tuiles

Un raster peut être découpé en plusieurs « sous-raster » selon ses dimensions spatiales (rangées et colonnes). Ces tableaux⁴⁴ sont appelés tuiles et chacune de ces tuiles est définie sur un sous-ensemble du domaine spatial de la couverture du raster. L'avantage de cette technique est de pouvoir travailler sur plusieurs petits tableaux plutôt que sur un seul grand tableau. En particulier, les traitements ciblés sur une région particulière du raster nécessitent uniquement le balayage des pixels associés aux tuiles concernées par l'opération. Par contre, cette technique a pour inconvénient de multiplier les effets de bords⁴⁵ à la limite du raster (en l'occurrence : de la tuile).

La définition du tuilage diffère sensiblement d'un SGBD à l'autre. PostGIS (Racine & Cumming, 2011) propose un tuilage très souple (Figure 19) où les tuiles peuvent être irrégulières (Figure 19b et Figure 19f) ou régulières (Figure 19c et Figure 19d). Elles peuvent se recouvrir (Figure 19a) et des tuiles peuvent être non-définies (Figure 19b, Figure 19c, Figure 19f). Du côté d'Oracle (2003), le tuilage, tel que représenté sur les Figure 19d et Figure 19e, est obligatoirement :

- strict (les tuiles ne se recouvrent pas),
- régulier (les tuiles ont toutes la même taille),
- complet (défini sur l'ensemble du raster).

Cette importante différence entre ces deux SGBD provient du fait que PostGIS ne fait pas de distinction entre une tuile et un raster : « *un raster est une tuile et une tuile est un raster* ». Comme nous le verrons dans la section 2.4, bien qu'à première vue, PostGIS semble proposer un tuilage plus polyvalent qu'Oracle, ce dernier engendre beaucoup moins de redondance de l'information et est donc plus performant dans le cas du tuilage régulier, strict et complet.

⁴⁴ Rappel : un raster est un tableau.

⁴⁵ La relation entre proximité des cellules et proximité des points de l'espace géographique n'est que partiellement conservée sur les bords du raster ou de la tuile.

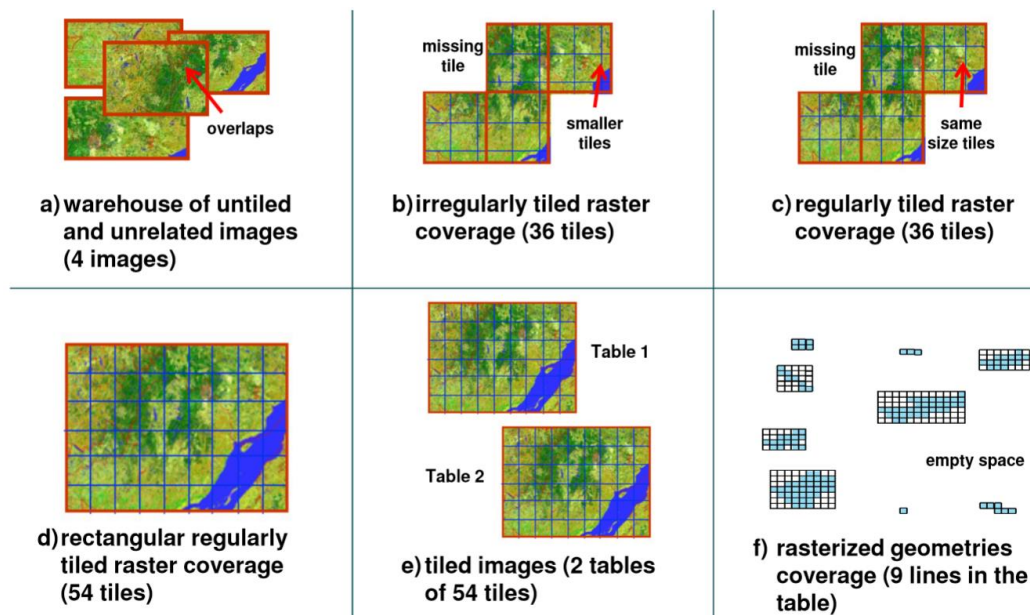


Figure 19. Les différents tuilages raster proposés dans PostGIS (Racine & Cumming, 2011)

Notons que bien que la notion de tuile (ou bloc) soit généralement utilisée pour un découpage selon les dimensions spatiales, un raster peut également être découpé selon sa dimension « bande ». De la même façon, cette approche permet la manipulation de raster de taille réduite par rapport au raster accueillant toutes les bandes de la couverture.

Enfin, soulignons le fait que le tuilage permis par Oracle (2003) est utilisé de manière similaire dans les cubes de données MOLAP (voir section 1.3.5.2).

2.3.2.2. Pyramides

Le changement de l'échelle d'analyse peut conduire à une optimisation des traitements en modifiant la résolution spatiale des cellules. Ainsi, la plupart des SGBD spatiaux (Oracle, 2003 ; Racine & Cumming, 2011) et des SIG-logiciels (ESRI, 2012) autorisant la manipulation des données raster proposent le stockage de plusieurs tuiles de taille constante sur toute la couverture raster (ou couverture matricielle) à différents niveaux de résolution spatiale (Figure 20). Ces tuiles sont hiérarchisées comme une simple dimension d'un OLAP en fonction de leur emprise spatiale. Cette hiérarchie est toujours stricte, symétrique, et généralement couvrante, unique et statique (voir section 1.2.1).

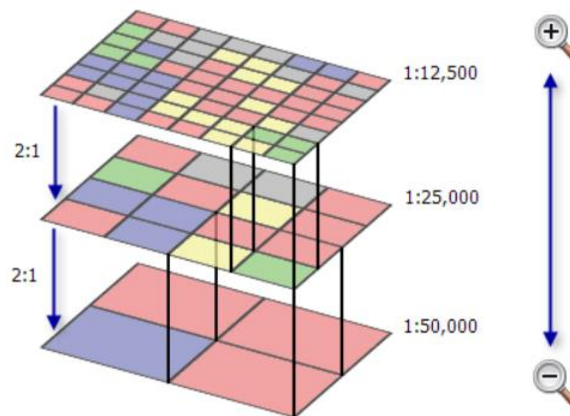


Figure 20. Exemple de pyramide raster à trois niveaux (ESRI, 2012)

A partir d'un raster de base et de différents niveaux d'échelles définis par l'utilisateur, le SIG construit les raster qui constituent les niveaux de la pyramide (Adelson *et al*, 1984). La plupart du temps, les niveaux construits sont définis à une échelle inférieure au raster de base et sont calculés par des méthodes d'agrégation⁴⁶ (ESRI, 2012), en réduisant le nombre de rangées et de colonnes, et en diminuant la résolution spatiale. Le choix de la méthode diffère en fonction de la signification sémantique des cellules. Grâce à cette technique, les traitements nécessitant une échelle d'analyse inférieure à l'échelle du raster de base (typiquement, la fonction « zoom out » d'une interface cartographique) sont optimisés puisque les raster manipulés sont plus petits (et plus généralisés).

Encore une fois, nous pouvons remarquer une forte similitude avec une technique d'optimisation des requêtes de l'OLAP : la matérialisation de cuboïdes de données (voir section 1.2.5).

2.4. Modèles physiques de stockage raster

Il existe plusieurs stratégies pour le stockage physique de raster dans un SGBD. Les deux plus connues sont les modèles GeoRaster d'Oracle (2003) et plus récemment, PostGIS Raster (Racine & Cumming, 2011; Regina & Leo, 2011). Dans les prochaines sections, nous allons analyser ces deux modèles et en retirer les avantages respectifs.

2.4.1. Oracle GeoRaster

Oracle (2003) propose un stockage des raster à l'aide de deux objets⁴⁷ : SDO_GEORASTER et SDO_RASTER (Figure 21). SDO_GEORASTER est un type d'attribut autorisant le stockage d'une couverture spatiale raster. Dans une même table, cet attribut particulier peut ainsi être accompagné d'autres attributs de tous types définis par l'utilisateur (par exemple, un identifiant et un nom pour la couverture). Une table comprenant un attribut SDO_GEORASTER est appelée table GEORASTER. Un attribut SDO_GEORASTER est un objet comprenant lui-même plusieurs attributs et qui renvoie

⁴⁶ Par exemple : moyenne pour du quantitatif, pixel majoritaire pour du qualitatif (Eastman, 2009).

⁴⁷ Terminologie utilisée par Oracle, à ne pas confondre avec la notion d'objet dans une base de données orientée objet.

vers un ou plusieurs enregistrements qui sont définis comme des objets SDO_RASTER et stockés dans une table à part : la table RASTER DATA. Alors que l'objet SDO_GEORASTER ne stocke que des métadonnées, les enregistrements de la table RASTER DATA (SDO_RASTER) contiennent la définition de raster au niveau de la cellule organisés en blocs (un bloc par enregistrement).

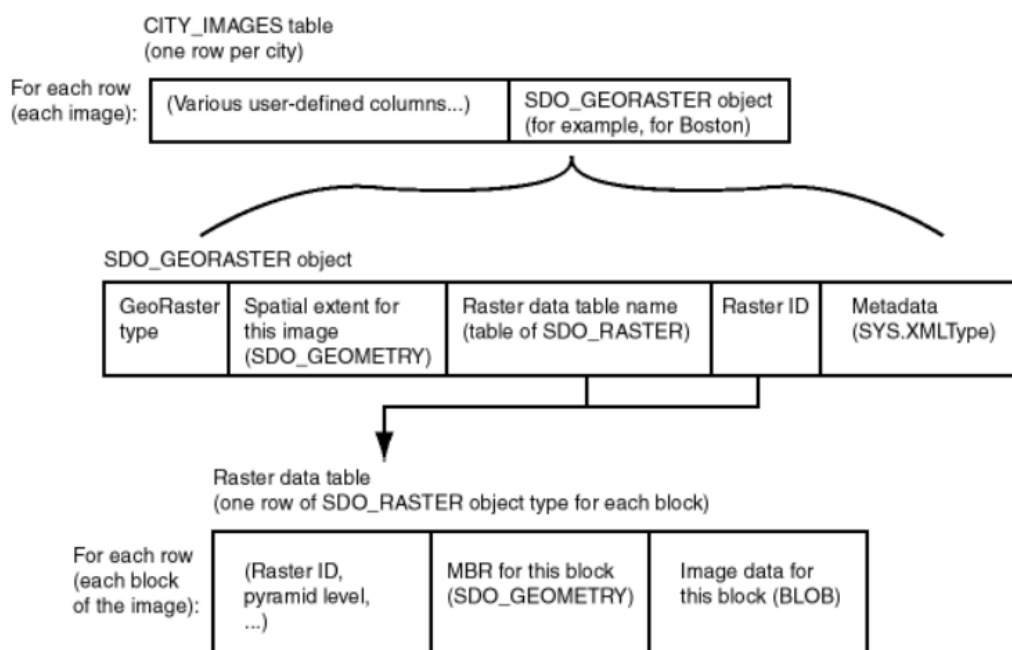


Figure 21. Stockage physique d'un raster dans Oracle GeoRaster (Oracle, 2003)

2.4.1.1. Objet SDO_GEORASTER

Un objet SDO_GEORASTER comprend un attribut « Metadata » définissant l'ensemble des métadonnées du raster au format XML. Ces métadonnées comprennent :

- Les informations sur l'objet : identifiant défini par l'utilisateur, description, version, etc.
- Les informations sur le raster : le type de valeur des cellules (binaire codé sur 1 bit, entier signé codé sur 8 bits, réel codé sur 32 bits, etc.), la taille des blocs en pixels, le type d'entrelacement⁴⁸ (BSQ, BIL et BIP), la dimensionnalité, le type de compression et les informations sur la pyramide
- Les informations sur le système de coordonnées de référence pour passer de l'espace raster à l'espace géographique
- Des informations sur le système de référence temporel
- Des informations spectrales (bandes) pour les images géographiques
- Des informations sur chaque couche⁴⁹ définie dans l'objet raster : palette de couleurs, valeur « no data », histogramme, etc.

⁴⁸ Voir section 5.10.

⁴⁹ Oracle (2014) définit la bande et la couche respectivement comme des concepts physiques et logiques. En général, une couche est physiquement stockée dans une bande.

A l'exception des informations sur l'objet et sur le raster, toutes les autres métadonnées sont optionnelles. Ainsi, un objet SDO_GEORASTER est aussi utilisable pour la gestion de tableaux multidimensionnels natifs (sans propriétés spatiales).

En plus de l'attribut « métadonnées », l'emprise spatiale de la couverture est dupliquée dans un attribut spécifique au format vecteur afin d'optimiser les requêtes spatiales par des méthodes d'indexation telles que le B-tree, R-tree (Kothuri *et al*, 2002), *etc.* Les attributs « Raster ID » et « Raster data table name » identifient de manière unique un ou plusieurs objets SDO_RASTER situés dans une table séparée (table RASTER DATA). Ainsi, une couverture spatiale peut être subdivisée en plusieurs blocs (ou tuiles) aux dimensions identiques en fonction de la rangée, la colonne, mais aussi la bande. Notons qu'un objet SDO_RASTER peut éventuellement être en relation avec plusieurs objets SDO_GEORASTER (cardinalité N-N).

2.4.1.2. Objet SDO_RASTER

Un objet SDO_RASTER est un bloc en relation avec un objet SDO_GEORASTER. Il comprend une série d'attributs permettant d'identifier son rôle dans l'objet SDO_GEORASTER : identifiant unique, niveau de pyramide, et son numéro de bloc au niveau de la rangée, la colonne et la bande. L'attribut MBR (« Minimum Bounding Rectangle ») définit l'emprise spatiale du bloc pour des raisons d'indexation spatiale comme pour la couverture.

L'attribut BLOB (« Binary Large Object ») renvoie vers un tableau multidimensionnel qui comprend toutes les valeurs de cellule du bloc. Celui-ci est interprété géométriquement et sémantiquement par le SGBD grâce aux métadonnées de la couverture à laquelle il se rapporte.

Notons qu'Oracle définit aussi la notion de raster vide. Il s'agit d'un bloc pour lequel tous les pixels ont la même valeur. Ce type de bloc particulier occupe une entrée dans la table RASTER DATA, mais le BLOB associé est de longueur nulle. La valeur de pixel récurrente est définie dans les métadonnées de la couverture. Cette méthode permet à la fois d'économiser l'espace de stockage et d'optimiser les traitements sur le bloc en question.

2.4.1.3. Evaluation

Bien qu'assez complexe, le modèle physique d'Oracle GeoRaster offre un stockage du raster très complet et optimisé pour le tuilage et les pyramides. En effet, pour chaque tuile définie dans la table RASTER DATA, seuls le tableau multidimensionnel (BLOB) et ses identifiants (bloc, pyramide et identifiant absolu) sont stockés. Toutes les métadonnées nécessaires à l'interprétation géométrique et sémantique du BLOB sont rassemblées dans l'objet SDO_GEORASTER qui se rapporte à la couverture spatiale. En revanche, cette structure n'offre pas une totale liberté dans le sens où le tuilage doit être géométriquement régulier et stricte (pas de recouvrement entre les tuiles qui doivent être de même taille).

2.4.2. PostGIS Raster

PostGIS (Racine & Cumming, 2011; Regina & Leo, 2011) propose un modèle de stockage raster sensiblement différent d'Oracle. Ici, la philosophie n'est pas d'optimiser les traitements et l'espace de stockage, mais plutôt d'offrir une grande liberté à l'utilisateur à travers un modèle de données relativement simple et autorisant des opérations mêlant données raster et vectorielles.

2.4.2.1. Type d'attribut « raster »

Le modèle raster de PostGIS (Figure 22) est assez similaire à son modèle vecteur : tout comme un enregistrement peut être associé à une entité vectorielle au moyen d'un attribut de type « geometry », celui-ci peut aussi être associé à un raster au moyen d'un attribut de type « raster ». Un attribut « raster » contient non seulement les données brutes du raster sous forme de tableau (équivalent du BLOB chez Oracle), mais également toutes les métadonnées nécessaires à son interprétation par le SGBD : paramètres de géoréférenciation, nombre de rangées et de colonnes, nombre de bandes, *etc.* PostGIS offre aussi la possibilité de considérer l'attribut raster comme un pointeur vers un fichier stocké en dehors de la base de données.

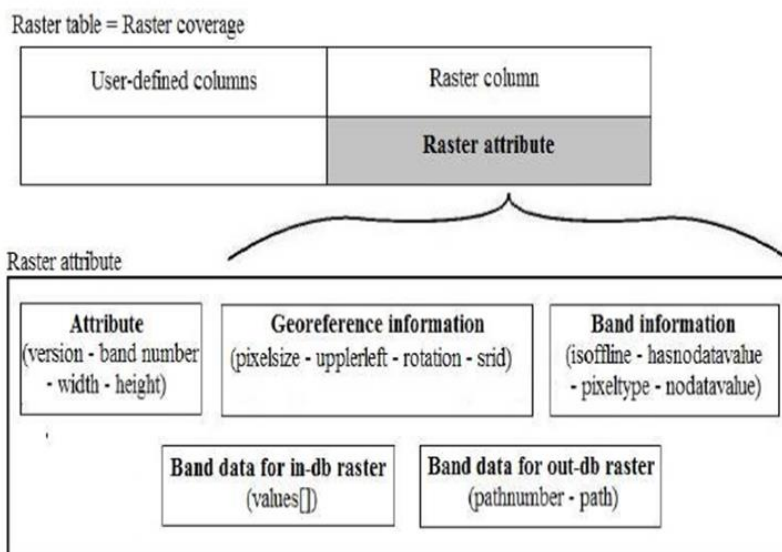


Figure 22. Stockage physique d'un raster dans PostGIS (Dang, 2012)

2.4.2.2. Table « raster_columns »

Tout comme la metatable⁵⁰ « geometry_columns » pour le vecteur, la metatable « raster_columns » permet d'imposer une série de contraintes sur un attribut « raster » :

- le système de coordonnées de références (SRID)
- la possibilité ou non de recouvrement entre raster
- l'obligation ou non d'alignement entre raster
- la résolution spatiale
- le nombre de rangées, colonnes, bandes
- le codage des pixels
- la valeur « no data »
- l'emprise spatiale (sous forme d'objet de type « geometry »).

2.4.2.3. Evaluation

Cette simple structure offre à l'utilisateur la possibilité d'organiser le tuilage de ses raster comme bon lui semble puisque les notions de tuile (ou bloc) et de raster sont ici confondues (voir section 2.3.2.1). En revanche, même si toutes les tuiles partagent les mêmes métadonnées, celles-ci doivent être répétées pour chacune d'entre elles (ce qui n'est pas le cas dans Oracle). Enfin, grâce à la similitude entre attribut de types raster et vecteur, il est très facile d'appliquer des traitements mélangeant ces deux formats. PostGIS propose d'ailleurs plusieurs fonctions dont la syntaxe est identique pour les paramètres raster et vecteur.

2.5. Conclusion

Au cours de la première partie de ce chapitre, nous avons défini les concepts de base du SIG en mettant l'accent sur deux dualités : d'un côté les données vectorielles et les données raster (dont la structure physique présente beaucoup de similarités avec un cube de données MOLAP), de l'autre côté les phénomènes spatialement continus et spatialement discrets. Sur base des caractéristiques respectives du mode vecteur et raster, un bilan a été dressé sur leur application en fonction du caractère discret ou continu d'un phénomène géographique. En particulier, les phénomènes continus semblent très adaptés au raster lorsqu'ils sont modélisés comme des champs réguliers en raison de la gestion implicite de la géométrie et de la topologie. A l'inverse, pour les champs irréguliers, le mode vecteur semble plus adapté en raison de sa souplesse au niveau géométrique. Nous avons ensuite abordé la notion de couverture spatiale telle que définie par l'ISO et l'OGC car elle constitue une norme solide pour la modélisation conceptuelle des phénomènes géographiques discrets et continus, indépendamment du type de modélisation (vecteur ou raster).

La deuxième partie du chapitre était centrée sur la modélisation raster qui constitue la base de notre hypothèse de recherche (voir section 4.3) en raison de toutes les similitudes qu'elle comporte avec les cubes MOLAP. Quelques caractéristiques

⁵⁰ En réalité, il ne s'agit pas d'une table, mais d'une vue depuis la version 2 de PostGIS.

importantes ont ainsi été définies: l'algèbre de carte, les tuiles et les pyramides. Pour terminer, deux SGBD spatiaux capables de stocker et manipuler des données raster ont été présentés et comparés : Oracle et PostGIS. Cette comparaison a révélé deux modèles physiques aux philosophies sensiblement différentes. D'un côté, Oracle optimise le stockage des tuiles et leur traitement à travers une architecture relativement complexe. De l'autre côté, PostGIS propose une forte interopérabilité entre données raster et vecteur à travers une architecture plus simple et plus souple.

Chapitre 3 : SOLAP

L'OLAP est apparu pour combler une importante lacune des systèmes OLTP, à savoir leur incapacité à fournir une analyse décisionnelle efficace (voir chapitre 1). Puisque les SIG ont été pensés à l'origine comme des outils transactionnels, ils souffrent exactement du même problème et ne peuvent donc pas répondre aux exigences d'un véritable système décisionnel (Proulx *et al*, 2004) : multidimensionnalité, réponses synthétiques, rapidité, *etc.* D'un autre côté, les systèmes OLAP à eux seuls sont incapables d'exploiter la composante spatiale⁵¹ des données à la manière d'un SIG : interface cartographique, stockage des géométries, analyse spatiale, *etc.* C'est pourquoi Bédard (1997), en associant les technologies OLAP et SIG, a donné naissance à l'OLAP spatial (SOLAP) : « *une plateforme visuelle conçue spécialement pour supporter une analyse spatio-temporelle rapide et efficace à travers une approche multidimensionnelle qui comprend des niveaux d'agrégation cartographiques, graphiques et tabulaires* ». De cette manière, la composante SIG présente dans le SOLAP enrichit l'analyse multidimensionnelle de l'utilisateur grâce à une interface cartographique, mais aussi grâce à l'intégration de requêtes capables d'exploiter le caractère spatial des données.

Puisqu'un SOLAP n'est jamais qu'un enrichissement d'un système décisionnel classique, tous les aspects vus dans le premier chapitre (informatique décisionnelle) restent d'application dans ce domaine. Ainsi, les trois éléments de l'architecture décisionnelle doivent être capables de gérer l'information spatiale : ETL, entrepôt de données et serveur SOLAP. En particulier, la spatialité des entités doit être clairement définie dans le schéma (étoile, flocon, *etc.*) de l'entrepôt de données⁵² et le serveur SOLAP doit être capable de fournir des requêtes spatiales tout en respectant les normes de performance instaurées par Codd et l'OLAP Report.

Théoriquement, puisqu'il est soumis à tous les aspects de l'informatique décisionnelle, un SOLAP devrait également être soumis à tous les aspects du SIG (en particulier les deux dualités vecteur/raster et discret/continu). En pratique, ce n'est pas (encore) le cas car les applications SOLAP présentes sur le marché utilisent exclusivement le mode vecteur et sont inaptes à la modélisation des phénomènes spatialement continus. Cet aspect important sera abordé plus en profondeur dans le chapitre 4.

Le présent chapitre a pour but d'énoncer les caractéristiques qui sont spécifiques au domaine du SOLAP. La section 3.1 sera consacrée à la spatialisation de la structure multidimensionnelle d'un entrepôt de données. Cette spatialisation implique une interface cartographique complémentaire aux tableaux et graphiques, ainsi qu'une redéfinition spatiale des opérations OLAP classiques (coupes spatiales, forages spatiaux, *etc.*). Ces deux aspects seront traités respectivement dans les sections 3.2 et 3.3. Avant de conclure, nous aborderons les deux types d'architecture SOLAP relationnelle et multidimensionnelle

⁵¹ Il a été estimé qu'environ 80% des données transactionnelles présentent un caractère spatial (Franklin, 1992).

⁵² On parle alors d'entrepôt de données spatial.

(section 3.4), ainsi que la classification des principaux outils SOLAP existants en fonction de leurs composantes respectives OLAP et SIG (section 3.5).

3.1. Spatialisation de l'entrepôt de données

Un OLAP devient spatial lorsqu'un ou plusieurs élément(s) de la structure multidimensionnelle de son entrepôt sont définis spatialement à la manière d'un SIG. Cette spatialisation a lieu sur les dimensions et/ou sur les mesures comme nous allons le décrire dans les prochaines sections. Un SOLAP est considéré comme tel s'il contient au moins une dimension spatiale⁵³ ou au moins une mesure spatiale.

3.1.1. Dimension spatiale

La manière la plus évidente (et la plus courante) d'intégrer la spatialité dans une structure multidimensionnelle est de l'associer à une ou plusieurs dimension(s), ou plus exactement à un ou plusieurs niveau(x) hiérarchique(s) de dimension. Pour spatialiser un niveau, il convient de lui associer un attribut complexe qui définit la géométrie de chacun de ses membres. Ainsi, les membres d'un niveau de dimension peuvent être représentés par des objets vectoriels et peuvent participer à des requêtes spatiales. Notons qu'il est tout à fait possible d'associer des entités vectorielles de différents types (points, lignes ou polygones) aux différents niveaux d'une même dimension spatiale. Par exemple, prenons un accident de voiture dont les niveaux de dimension spatiale sont :

- une localisation absolue représentée par un point (niveau le plus détaillé),
- une route représentée par une polyligne (niveau intermédiaire),
- une commune représentée par un polygone (niveau le moins détaillé).

Conceptuellement, une dimension peut aussi être considérée comme « spatiale » malgré l'absence de géométrie. Par exemple, il est possible de représenter des entités administratives géométriquement indéfinies dans une table de pivot ou un graphique affichant la mesure « densité de population », bien qu'il ne soit plus possible de représenter ces membres sur une carte ou de leur appliquer des requêtes spatiales. Afin de lever cette ambiguïté, Rivest *et al* (2003) utilisent les termes « dimension géométrique » pour une dimension spatiale dont les niveaux sont associés à une géométrie, « dimension non-géométrique » pour une dimension spatiale où aucun niveau n'est associé à une géométrie et « dimension mixte » pour une dimension spatiale comprenant les deux types de niveau (Figure 23).

⁵³ Dans la littérature SOLAP, le terme « dimension spatiale » englobe à la fois les caractéristiques spatiales et sémantiques des membres qui la constituent (par exemple : les noms des communes et leurs géométries). Le terme « dimension géographique » est également employé pour désigner ce type de dimension (Bimonte, 2007).

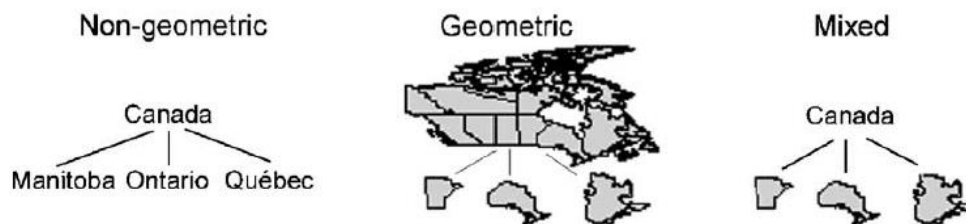


Figure 23. Les trois types de dimensions spatiales d'un outil SOLAP (Rivest et al, 2003)

Généralement, les liens entre les membres d'une dimension spatiale issus de différents niveaux hiérarchiques sont construits selon des relations « composé-composant » qui peuvent être représentées par des associations d'agrégation ou de composition dans un diagramme de classe UML. Cependant, la littérature nous montre que ce n'est pas toujours le cas. Par exemple, Marchand *et al* (2004) utilisent des dimensions d'opérateurs topologiques spatiaux pour l'analyse de déplacement de radios amateurs. Les membres de ces dimensions sont alors reliés par des relations de types « adjacence », « inclusion », « intersection », *etc.*

3.1.2. Mesure spatiale

Tout comme la dimension, la mesure d'un SOLAP peut acquérir des propriétés spatiales. Au lieu de contenir une valeur numérique comme on l'observe dans le cas d'une mesure issue d'un OLAP typique⁵⁴, une mesure spatiale contient un pointeur vers une ou plusieurs entités spatiales (Han *et al*, 1998). Il convient alors de clairement définir les méthodes d'agrégation de ces mesures grâce à des opérations de généralisation issues du domaine du SIG (l'union, l'intersection, la fusion, l'enveloppe convexe, *etc.*).

Intégrer la spatialité dans la mesure ou dans la dimension permet de formuler des requêtes SOLAP répondant à des questions fondamentalement différentes. Certaines modélisations SOLAP autorisent d'ailleurs la permutation des mesures et des dimensions (Han *et al*, 1998 ; Bimonte, 2007). Illustrons ce concept à l'aide de l'exemple de la section précédente. Si les entités administratives sont considérées comme des membres d'une dimension spatiale et la densité de population comme une mesure, le système répondra à des questions de type « Quelle est la densité de population des différentes provinces du Canada? ». Inversement, si la densité de population est considérée comme une dimension composée de membres associés à une classe de densité, et les entités administratives comme une mesure spatiale⁵⁵, le système répondra à des questions de type « Quelles sont les provinces du Canada ayant une densité de population comprise entre 4 et 6 habitants/km²? ».

Un autre type de mesure spatiale est également présent dans la littérature : les valeurs numériques exprimant le résultat d'opérateurs d'analyse spatiale (Malinowski et Zimányi, 2004 ; Rivest *et al*, 2005; Bimonte, 2007). Ceux-ci sont soit métriques (une distance, une surface, *etc.*), soit topologiques (nombre d'entités avec une frontière

⁵⁴ Nous avons vu à la section 1.2.2 que les mesures numériques étaient les plus utilisées dans les entrepôts de données.

⁵⁵ En considérant que la mesure est agrégée par l'union.

commune, nombre d'entités partiellement recouvertes, etc.). Par exemple, un modèle SOLAP peut inclure une mesure « distance cumulée des habitations par rapport à l'hôpital le plus proche » en fonction de la commune et du type de ménage (opérateur métrique).

Notons que la distinction entre mesures atomiques et mesures dérivées vue à la section 1.2.2 reste d'application avec les mesures spatiales. Une mesure spatiale dérivée résulte d'un traitement appliqué à la volée sur des membres spatiaux et/ou sur des mesures spatiales lors de l'analyse de l'utilisateur. Prenons par exemple les polygones générés par l'intersection de membres issus de deux niveaux spatiaux faisant partie de deux dimensions spatiales distinctes : les entités administratives et les aires d'influence d'antennes GSM.

3.2. Interface cartographique

Comme pour l'OLAP, un SOLAP doit être capable de jongler entre la table de pivot et les graphiques afin de livrer une information synthétique et complète à l'utilisateur, mais la véritable force du SOLAP réside dans la troisième interface qu'il propose : la carte.

Alors que les graphiques montrent très clairement l'évolution d'une ou plusieurs mesure(s) en fonction d'une, voire deux dimensions, et que la table de pivot offre la vue la plus précise et la plus riche en termes de nombre de dimensions et mesures, la carte est le moyen le plus évident et le plus efficace de représenter les dimensions et mesures spatiales. En effet, une carte révèle au premier coup d'œil les relations topologiques des objets spatiaux et les distances qui les séparent. De plus, les mesures de l'ED sont facilement mises en relation avec des entités géographiques provenant de couches cartographiques externes. Le graphique peut éventuellement être intégré à la carte pour augmenter le nombre de dimensions ou mesures visibles, bien que cela entraîne une réduction de la vision synoptique de l'utilisateur (Figure 24).

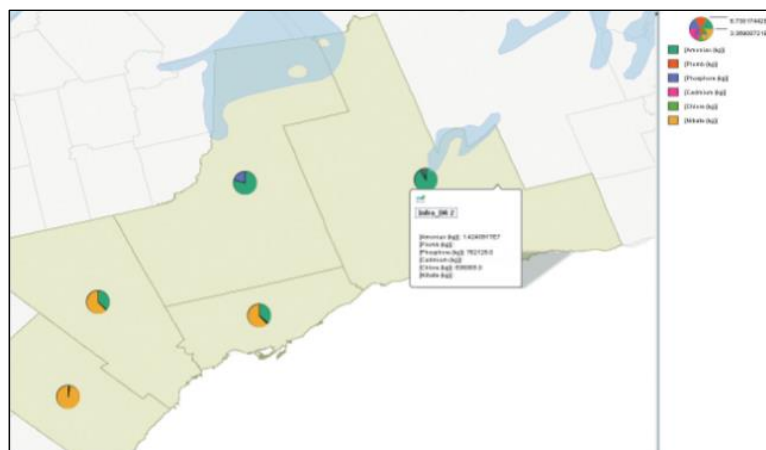


Figure 24. Exemple d'intégration de graphiques dans une interface cartographique SOLAP dans Map4Decision (Intelli³, 2010). D'après Tranchant (2013).

L'interface cartographique d'un SOLAP (Figure 25) doit permettre à l'utilisateur de voyager dans le cube de données de manière interactive et intuitive tout en respectant les règles de sémiologie cartographique, en particulier celles concernées par la cartographie thématique (Bertin, 1967 ; Denègre, 2005 ; Donnay, 2013). Ces règles doivent être

implémentées de telle sorte qu'elles soient quasiment transparentes pour l'utilisateur qui, la plupart du temps, n'est pas sensibilisé aux problèmes engendrés par les données spatiales : représentation des données, géoréférencement, qualité géométrique des données, etc. De plus, les opérations SOLAP appliquées sur la carte doivent être mises à jour simultanément sur les interfaces tabulaires et graphiques selon les mêmes règles sémiologiques afin de conserver « une continuité perceptuelle nécessaire à la découverte de corrélations » (Bédard et al, 2005).

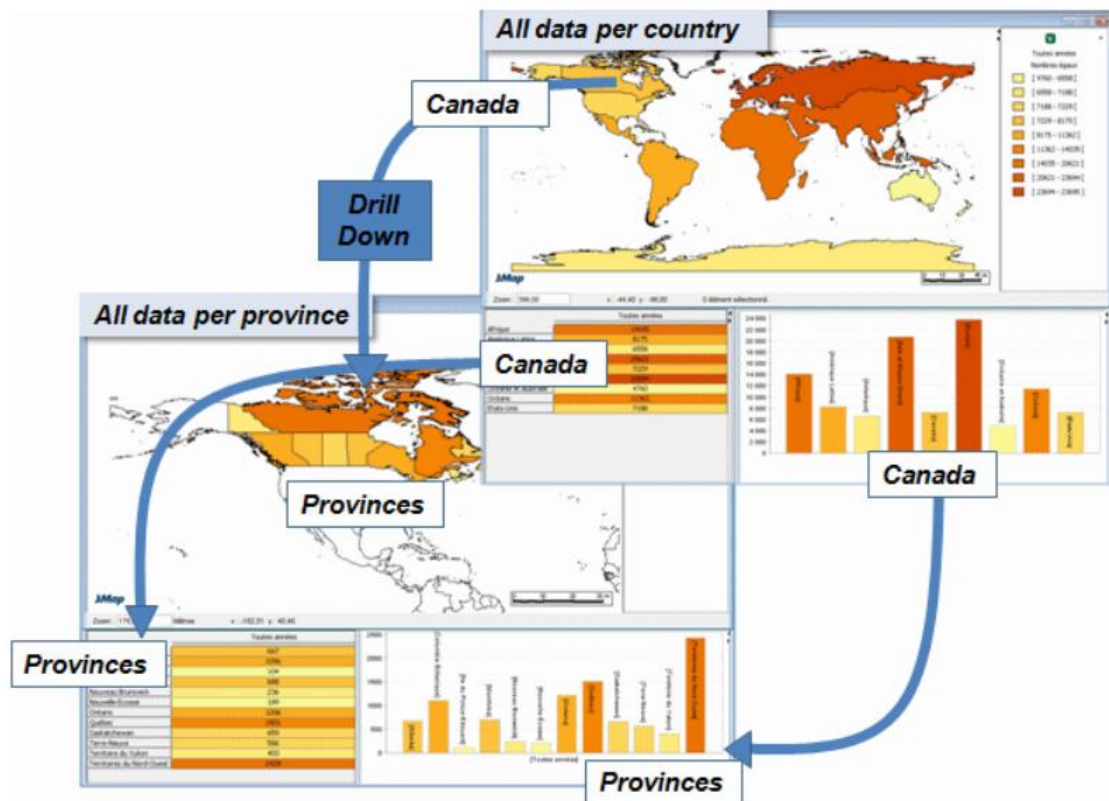


Figure 25. Exemple d'opération « drill down » avec l'interface de JMAP, maintenant devenu Map4Decision (Intelli³, 2010). D'après Bédard et al (2009).

3.3. Opérations SOLAP

Jusqu'à présent, nous avons vu l'impact de l'enrichissement spatial de l'OLAP sur la structure multidimensionnelle de l'ED et sur les interfaces. Le serveur SOLAP, à qui revient la tâche d'interpréter le cube de données via la structure multidimensionnelle de l'ED (du moins, dans une architecture ROLAP ou HOLAP), nécessite une redéfinition de ses opérations telles que décrites à la section 1.3.2. Lorsqu'elles concernent une dimension spatiale, on parle alors de forages spatiaux (« spatial drill down », « spatial roll up ») et de coupes spatiales (« spatial slice »). Ces opérations peuvent être exécutées interactivement, en cliquant directement sur les éléments de la carte.

La Figure 25 montre une opération « spatial drill down » sur le membre « Canada » de la dimension spatiale « entités administratives » du niveau « pays ». Ce forage est opéré par un simple clic sur le membre de la carte que l'utilisateur désire investiguer (ici, le Canada). Le résultat du « drill down » est une carte affichant les données désagrégées selon

les membres « enfants » (niveau inférieur) du membre « Canada » : les provinces du Canada. La Figure 25 montre aussi l'opération « roll up » si on inverse le sens des flèches représentées, à savoir le passage du niveau « provinces » au niveau « pays ». Un « spatial roll up » sur le membre « all » de la dimension « entités administratives » éliminerait cette dimension spatiale de l'analyse, ce qui n'a pas de sens dans l'interface cartographique (en admettant que le schéma de l'ED ne comprenne qu'une dimension spatiale), mais bien dans les interfaces graphique et tabulaire (table de pivot).

Toujours à partir de la Figure 25, on imagine aisément un « spatial slice » où l'utilisateur désirerait afficher les données agrégées seulement selon les entités administratives de certains pays (par exemple, le Canada et les États-Unis).

Bien sûr, les opérations classiques de l'OLAP sont applicables sur des dimensions non-spatiales avec un impact sur l'interface cartographique. En fixant le niveau de la dimension spatiale à « pays », l'utilisateur a par exemple la possibilité d'effectuer des forages ou des coupes temporel(le)s pour analyser les données à des époques particulières.

En dehors de leur interactivité et de leur représentation avec la carte, les opérations SOLAP appliquées sur des dimensions spatiales ne sont donc pas très différentes des opérations OLAP. Par contre, les opérations SOLAP appliquées sur des mesures spatiales nécessitent une définition des fonctions d'agrégations spécifiques à la géométrie des mesures (points, lignes, polygones) et à leur signification sémantique.

3.4. Architectures SOLAP

La plupart des SOLAP sont implémentés en ROLAP car cette structure s'apparente davantage à celle d'un SIG vectoriel que dans le cas du MOLAP. De plus, aucun outil MOLAP n'est capable de directement stocker des géométries, contrairement aux SGBD relationnels sur lesquels s'appuient les ROLAP.

Dans un SOLAP relationnel, la spatialisation a lieu en définissant un attribut complexe de type « géométrie » à une table de dimension dans le cas d'une dimension spatiale, ou à la table des faits dans le cas de mesures spatiales ou de dimensions spatiales dégénérées (voir section 1.2.4). GeoMondrian (Spatialytics, 2014) permet ainsi de spatialiser les dimensions du serveur Mondrian (Pentaho, 2014).

Dans le cas d'un SOLAP multidimensionnel, la solution est de confier la gestion des géométries à un outil séparé (serveur SOLAP). Le cube MOLAP contient alors des pointeurs renvoyant vers des géométries gérées par le serveur SOLAP. C'est le cas de Map4Decision (Intelli³, 2010) qui est ainsi capable de se connecter à des entrepôts de données aussi bien relationnels que multidimensionnels.

3.5. Classification des outils SOLAP

Depuis sa création, le SOLAP a fait l'objet de nombreuses recherches et est maintenant implémenté dans une quarantaine d'applications commerciales et quelques produits « open source ». Toutes ces solutions rentrent dans une classification en fonction de l'importance relative de leurs composantes SIG et OLAP : OLAP-dominant, SIG-

dominant et OLAP-SIG intégré (Bédard *et al*, 2005; Proulx, 2009; Bédard, 2014). Cette section a pour but de décrire ces différentes catégories.

3.5.1. OLAP dominant

Un outil OLAP-dominant offre toutes les fonctionnalités que l'on attend d'un OLAP : structure multidimensionnelle, entrepôt de données relationnel ou multidimensionnel, interfaces tabulaire et graphique interagissant l'une avec l'autre, rapidité des requêtes, gestion transparente de cuboïdes de données, *etc.*

Le volet SIG est lui aussi présent mais de manière beaucoup plus limitée que l'OLAP. L'interface cartographique offre des opérations de base de navigation (déplacement, changement d'échelle, *etc.*) et de sélection des éléments géométriques. Par contre, les opérations typiquement attribuées au SOLAP telles que le forage spatial ne sont pas toujours implémentées et les opérations d'analyse spatiale sont inexistantes. De plus, les cartes ne sont pas modifiables sémiologiquement (par exemple : classification thématique imposée par le logiciel) et sont limitées à une seule dimension spatiale. Généralement, une dimension spatiale ne peut être construite qu'avec une ou deux primitives vectorielles (points, polygones et très rarement les lignes). Les entités complexes (multipoints, multilignes, *etc.*) sont absentes de cette catégorie. Enfin, les interactions entre la carte et les autres interfaces (table et graphique) sont rares et limitées.

Un outil OLAP dominant est donc un OLAP auquel quelques fonctionnalités cartographiques ont été rajoutées. La Figure 26 montre l'interface cartographique d'une solution OLAP-dominant : Oracle BI Server combiné à Oracle Java Viewer (Oracle, 2015).



Figure 26. Exemple d'application OLAP-dominant : Oracle BI Server + Oracle Java Viewer (Oracle, 2015). D'après Proulx (2009).

3.5.2. SIG dominant

Un outil SIG-dominant est toujours basé sur un ED relationnel (Bédard *et al*, 2005) et est donc également considéré comme un ROLAP. Ainsi, il est possible de programmer la plupart des fonctionnalités d'un OLAP en utilisant les propriétés relationnelles du SGBD, généralement à travers le langage SQL. Cependant, nous avons vu que ce langage

n'est pas aussi adapté à ce genre de programmation que le langage MDX qui est typique des cubes de données physiques⁵⁶ (MOLAP). Les capacités OLAP s'en retrouvent donc limitées. De plus, un SIG n'offre pas forcément toutes les interfaces d'un OLAP (table de pivot et graphiques).

Par contre, ce type d'outil offre un grand nombre d'opérations propres au domaine du SIG : navigation dans la carte, analyse spatiale, sémiologie cartographique, gestion de tous les types de géométries, gestion de plusieurs dimensions spatiales, etc.

Un outil SIG-dominant est donc un SIG programmé pour répondre à des fonctionnalités OLAP. La Figure 27 montre l'interface cartographique d'une application SIG-dominant d'après Proulx (2009) : le serveur OLAP Business Objets, maintenant racheté par SAP (2015), combiné à la librairie .Net (Microsoft, 2014b) MapXtreme (MapInfo, 2015).



Figure 27. Exemple d'application SIG-dominant : le serveur OLAP Business Objets (SAP, 2015) combiné à MapXtreme (MapInfo, 2015). D'après Proulx (2009).

3.5.3. OLAP-SIG intégré

Idéalement, les solutions appartenant à cette famille devraient intégrer toutes les fonctionnalités de l'OLAP et du SIG. Avec les outils de cette catégorie, l'utilisateur peut facilement forer sur une carte afin de représenter spatialement toutes les combinaisons possibles des dimensions et mesures du cube. Par exemple, le forage sur un seul membre cartographique est une opération qu'on ne retrouve pas souvent dans les outils des deux autres familles. De plus, la carte doit être modifiable d'un point de vue sémiologique.

Des opérateurs d'analyse spatiale viennent enrichir l'analyse multidimensionnelle et les actions sur la carte ont des répercussions directes sur les interfaces OLAP : tableau et carte (et *vice versa*). L'outil peut être aussi bien ROLAP que MOLAP, sans restrictions sur le nombre de dimensions spatiales ou sur les types de primitives vectorielles utilisables. Il est aussi possible d'afficher plusieurs cartes en même temps pour retracer, par exemple, l'évolution temporelle d'une dimension spatiale.

⁵⁶ A l'exception de GeoMondrian (Pentaho, 2014 ; Spatialytics, 2014) puisque ce dernier intègre le MDX dans une architecture ROLAP (voir section 1.3.5.1).

Selon Bimonte (2007), ce type d'outil devrait également être capable de gérer les hiérarchies complexes (non-strictes, non-couvrantes, multiples, *etc.*). Ce dernier va même plus loin dans l'intégration de l'OLAP et du SIG en définissant un modèle « OLAP géographique ». La particularité de ce modèle est la gestion séparée de la composante sémantique et géométrique des entités géographiques qui définissent les mesures et les membres de dimension. Les concepts de mesures spatiales et dimensions spatiales sont alors redéfinis en « mesures géographiques » et « dimensions géographiques ». Une dimension géographique peut être caractérisée à la fois par :

- une « hiérarchie descriptive » en fonction de la description sémantique des membres géographiques (Figure 28a),
- une « hiérarchie spatiale » en fonction des relations topologiques d'inclusion et d'intersection des membres géographiques (Figure 28b),
- une « hiérarchie de généralisation » en fonction du niveau d'échelle d'analyse (Figure 28c et section 4.2.4).

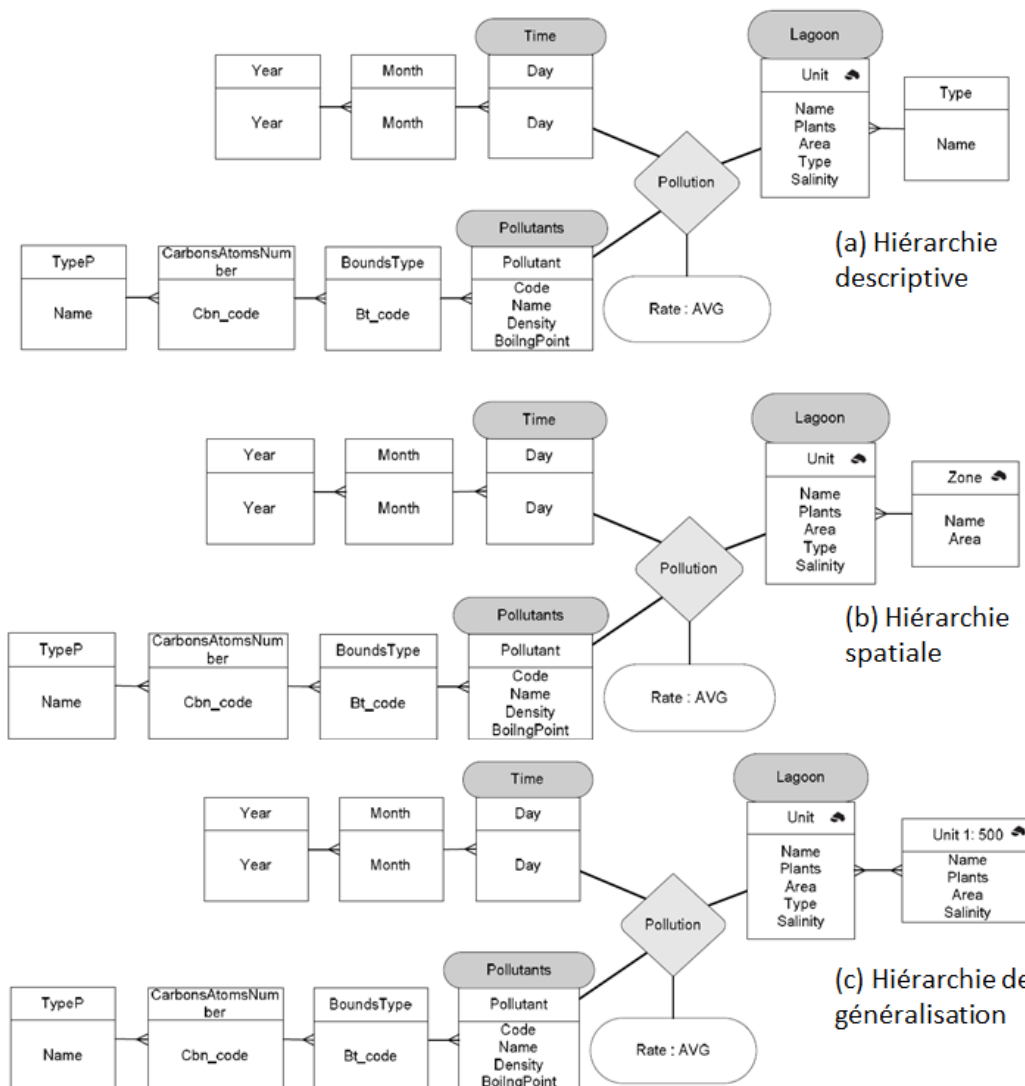


Figure 28. Les trois types de hiérarchies d'une dimension géographique (Bimonte, 2007)

Toujours dans le modèle « OLAP géographique », les mesures et les dimensions sont permutable, et les mesures géographiques peuvent elles aussi être caractérisées par les trois types de hiérarchies définis précédemment. Les mesures géographiques sont agrégées par des méthodes spécifiques simultanément pour les géométries et les valeurs sémantiques. Enfin, les opérateurs d'analyse spatiale permettent de modifier dynamiquement la structure multidimensionnelle du cube de données. Par exemple, un membre géographique créé à la volée par une opération de « buffer » peut être forcé afin d'agréger les données en fonction de ce nouveau membre.

En résumé, l'OLAP-SIG intégré représente l'outil SOLAP idéal puisque toutes les fonctionnalités OLAP et SIG sont présentes et interagissent les unes avec les autres. La Figure 24 et la Figure 25 montrent l'interface de Map4Decision (Intelli³, 2010) qui est actuellement considéré comme l'outil OLAP-SIG intégré le plus complet selon Bédard *et al* (2005), Proulx (2009) et Bédard (2014)⁵⁷. Le gros avantage de Map4Decision réside dans le fait qu'il soit directement pensé comme un outil OLAP-SIG intégré et qu'il nécessite donc très peu d'efforts de programmation. Il est théoriquement possible d'obtenir un OLAP-SIG intégré avec d'autres outils, notamment dans le domaine « open source », mais cela nécessite beaucoup de temps et de bonnes connaissances en informatique, en particulier en informatique décisionnelle et en SIG. Le serveur GeoMondrian (Spatialytics, 2014) est un outil clé⁵⁸ de ce genre d'architecture « open source » (même s'il ne gère que le ROLAP) puisqu'il intègre directement le spatial dans sa syntaxe MDX. Ce dernier est d'ailleurs compris dans la suite BI propriétaire Spatialytics (Spatialytics, 2014).

3.5.4. Bilan

Cette classification fonctionne bien pour comprendre les différentes approches dans la conception d'un SOLAP, mais elle ne devrait pas être là pour dénigrer les outils faisant partie des catégories OLAP-dominant et SIG-dominant. D'une part, le nombre de paramètres discriminants est trop important pour fixer une limite stricte entre un OLAP-dominant et un OLAP-SIG intégré ou entre un SIG-dominant et un OLAP-SIG intégré (d'autant plus que certains outils des catégories « dominant » présentent parfois des fonctionnalités intéressantes). D'autre part, le travail de Bimonte (2007) montre bien que les conditions d'un OLAP-SIG intégré ne sont pas forcément complètes et qu'elles peuvent évoluer. Ainsi, en enrichissant la définition de Bédard *et al* (2005), nous pourrions dire qu'il n'existe aucun outil OLAP-SIG intégré puisqu'aucun outil SOLAP ne permet la gestion des données raster qui représentent pourtant un volet important du SIG.

Enfin, cette classification démontre qu'il existe deux grandes approches pour concevoir un SOLAP : soit l'approche OLAP-dominant où on intègre les fonctionnalités SIG à un outil OLAP existant (par exemple : Mondrian), soit l'approche SIG-dominant où on intègre des fonctionnalités OLAP à un outil SIG existant (par exemple : PostGIS). Le prototype développé dans cette thèse fait partie de cette dernière catégorie (voir chapitre 8).

⁵⁷ Précisons qu'Yvan Bédard était à la tête de l'équipe de développement de JMAP à l'Université Laval (Canada). Plusieurs chercheurs de cette équipe (dont Marie-Josée Proulx) ont ensuite formé Intelli³ qui a repris le développement et la commercialisation de JMAP, maintenant appelé Map4Decision.

⁵⁸ En dehors du fait que GeoMondrian fonctionne uniquement en ROLAP.

3.6. Conclusion

Ce chapitre nous a permis de présenter les concepts de base sur lesquels repose la technologie SOLAP. En résumé, un SOLAP est le résultat de l'association de l'OLAP (voir chapitre 1) et du SIG (voir chapitre 2). Ce type d'outil permet à un décideur d'analyser de grands jeux de données agrégés selon plusieurs niveaux de granularité en fonction d'un modèle multidimensionnel avec une attention particulière portée sur la composante spatiale. Cette analyse s'appuie sur des cartes, des graphiques et des tableaux dans une interface simple et interactive. Afin d'enrichir l'OLAP avec les techniques du SIG, la structure multidimensionnelle des entrepôts de données doit intégrer l'information spatiale (géométrique) dans ses dimensions et/ou ses mesures. Ainsi, les opérations OLAP, redéfinies en opérations SOLAP, peuvent être appliquées directement sur la carte et sont combinables avec les opérations d'analyse spatiale typiques du domaine du SIG.

La dernière section du chapitre portait sur la classification des différents outils SOLAP actuellement sur le marché (propriétaires et « open source »). Cette distinction est basée sur l'importance relative des composantes OLAP et SIG dans les SOLAP : OLAP dominant, SIG dominant et OLAP-SIG intégré. La littérature montre bien que la catégorie OLAP-SIG intégré représente un idéal vers lequel la technologie SOLAP (qui est encore très jeune puisqu'elle a moins de 20 ans) devrait se diriger afin de permettre à l'utilisateur des analyses toujours plus souples et plus riches au niveau spatial. Alors que l'information raster constitue une part très importante du SIG, celle-ci est pratiquement absente du domaine SOLAP qui devrait pourtant être capable d'intégrer tous les formats de données dans son analyse. De plus, l'information spatialement continue qui représente aussi une part importante du SIG (à un niveau plus conceptuel) est absente des solutions offertes aux utilisateurs. Le prochain chapitre, dans lequel nous définissons les hypothèses de ce travail, présente les différents travaux de recherche axés sur ces deux problématiques : le raster et le spatialement continu intégrés au SOLAP.

Chapitre 4 : hypothèse

L'objectif principal⁵⁹ de ce travail se base à la fois sur les caractéristiques de l'OLAP et du SIG. Nous avons vu à la section 3.5 qu'un outil SOLAP, et en particulier un outil OLAP-SIG intégré, devrait théoriquement rassembler toutes les propriétés principales du SIG et de l'OLAP dans un seul environnement dont le but est l'analyse décisionnelle.

Or, comme le montre la littérature, les solutions SOLAP présentes sur le marché, même les plus performantes, ne couvrent qu'une partie du domaine du SIG : le modèle vectoriel et l'information spatialement discrète au détriment du modèle raster et de l'information spatialement continue. Les solutions sont partielles aussi dans le domaine de l'OLAP puisque les outils SOLAP traitent tous l'information spatiale comme du ROLAP au détriment du MOLAP (et donc également du HOLAP). En effet, même si un outil comme Map4Decision (Intelli³, 2010) est capable d'associer des géométries à un cube géré par un ED multidimensionnel (MOLAP), la spatialité est rejetée en dehors du cube qui ne stocke que des pointeurs. Par conséquent, les entités spatiales bénéficient des performances du MOLAP uniquement pour leur composante sémantique et pas leur composante géométrique. La Figure 29 résume cette problématique en reprenant certains aspects qu'un outil SOLAP devrait idéalement être capable de gérer. Alors que les caractéristiques notées en rouge sont déjà implémentées dans les SOLAP actuels, celles notées en bleu ne sont pour le moment qu'au stade de la réflexion et de prototypes de recherche.

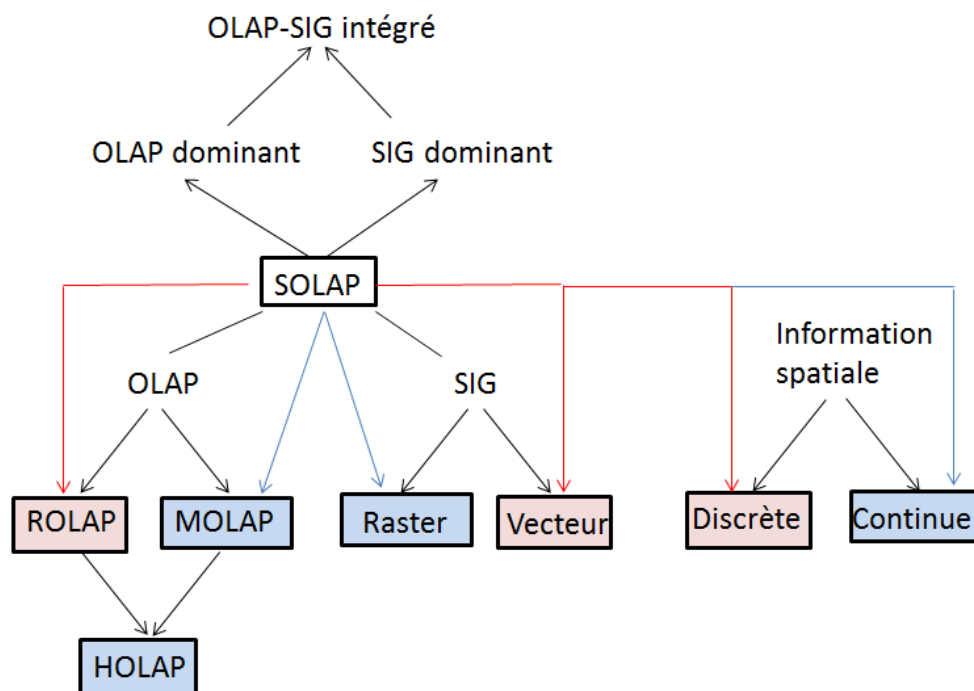


Figure 29. Caractéristiques de l'OLAP et du SIG théoriquement présentes dans le SOLAP

⁵⁹ Notre hypothèse de recherche se base avant tout sur notre objectif SOLAP (objectif principal). L'objectif « crime mapping », qui découle directement de l'objectif SOLAP, sera abordé dans les chapitres 7 et 8.

Ainsi, notre objectif de recherche qui portait sur l'enrichissement de la composante spatiale du SOLAP peut être reformulé par les trois questions suivantes:

- Comment intégrer le raster au SOLAP?
- Comment analyser l'information spatialement continue dans un SOLAP?
- Comment exploiter la géométrie des entités spatiales à l'intérieur d'un cube MOLAP?

La littérature scientifique apporte déjà plusieurs éléments de réponses aux deux premières questions⁶⁰. Après une analyse des quelques travaux traitant du SOLAP raster (maillé) et du SOLAP continu (« field OLAP »), nous formulerons notre hypothèse de recherche qui a pour ambition d'amener une réponse commune à ces trois questions.

4.1. SOLAP raster

4.1.1. Modèles SOLAP maillés

Les premiers travaux traitant de l'intégration de l'information géographique maillée dans le SOLAP proviennent de l'Université Laval. Il est important de préciser que dans ces travaux, nous ne pouvons pas toujours parler de raster au sens strict puisque ces modèles ont été implémentés en simulant un environnement raster à travers le mode vecteur de l'outil SOLAP Map4Decision. Nous faisons donc une différence entre le SOLAP raster exploitant les données raster sous forme de tableaux multidimensionnels, et le SOLAP maillé simulant l'empreinte géométrique de cellules raster avec des polygones vectoriels. Dans ce dernier cas de figure, une couverture spatiale est implémentée par un ensemble connexe de polygones de forme carrée, de taille régulière et disjoints. Cette importante distinction (raster/maillé) remet en question quelques-uns des concepts énoncés dans la littérature et rappelés dans les trois prochaines sections.

4.1.1.1. Miquel et al (2002)

Pour les besoins d'une application forestière, Miquel *et al* (2002) élaborent un modèle SOLAP où le niveau le plus fin de la dimension spatiale est une couche maillée. Puisque les membres des niveaux supérieurs de la dimension spatiale varient dans le temps, cette technique permet la comparaison de mesures sur des objets spatialement identiques entre différentes époques. En introduisant la notion de membre générique (invariable dans le temps) et de membre spécifique (variable dans le temps), le schéma en constellation (Figure 30) est constitué d'une table des faits génériques (associée aux dimensions génériques) et d'autant de tables des faits spécifiques (associées à des dimensions spécifiques) qu'il y a de fenêtres temporelles différentes. Ce travail démontre le potentiel du mode raster pour la gestion des dimensions spatiales dynamiques.

⁶⁰ Nous n'avons pas connaissance de travaux répondant directement à la troisième question, mis à part implicitement dans les travaux de McHugh (2008) et Plante (2014) présentés à la section 4.1.

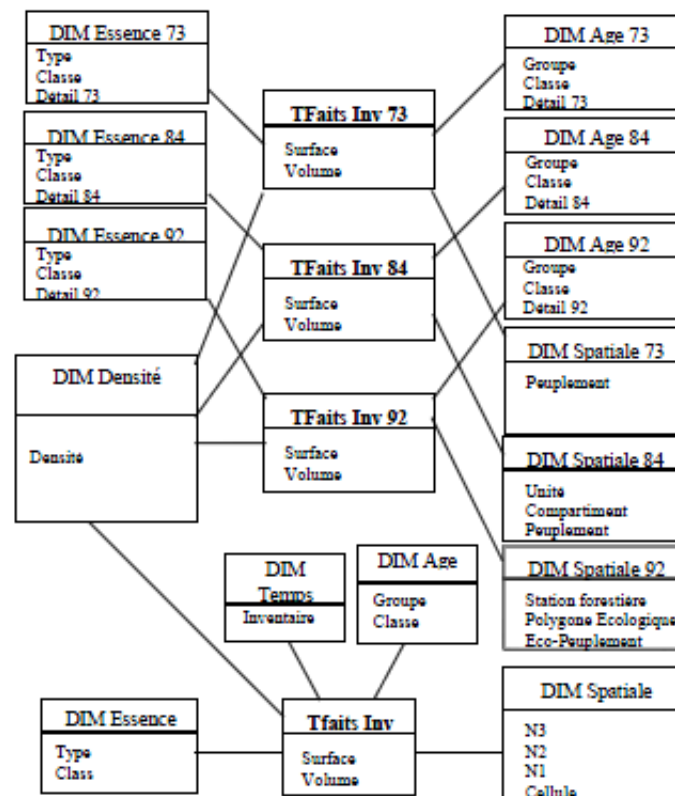


Figure 30. Schéma en constellation pour les inventaires forestiers (Miquel et al, 2002)

4.1.1.2. Mc Hugh (2008)

Dans McHugh *et al* (2007) et McHugh (2008), les premières bases théoriques de l'intégration de la structure matricielle (raster) dans le SOLAP sont posées. Cette structure peut ainsi être intégrée aux dimensions et aux faits.

Lorsqu'elle est intégrée à une dimension spatiale, une couche matricielle correspond à un niveau de dimension et une cellule correspond à un membre. Les trois types de dimensions spatiales mentionnés à la section 3.1.1 sont redéfinis en sept types différents, dont quatre nouveaux (Figure 31). Les quatre nouveaux types sont les dimensions matricielles (uniquement matricielles), mixtes matricielles (niveaux matriciels et niveaux non-spatiaux), hybrides (niveaux vectoriels et niveaux matriciels) et mixtes hybrides (niveaux vectoriels, matriciels et non-spatiaux). Ces dimensions peuvent être purement géométriques (amenant un découpage spatial pour l'agrégation des données, éventuellement à plusieurs niveaux de résolution) ou intégrer une composante thématique comme dans un SOLAP classique (par exemple : découpage selon les entités administratives). Notons que les dimensions hybrides amènent des problèmes de hiérarchies non-strictes entre niveaux matriciels et vectoriels lorsqu'ils sont reliés par une relation topologique d'inclusion.

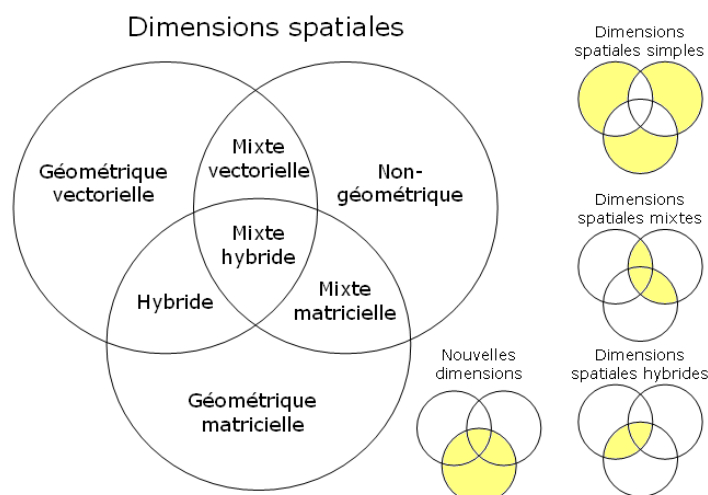


Figure 31. Les 7 types de dimensions spatiales en distinguant le vecteur et le matriciel (Mc Hugh, 2008)

Lorsque la structure matricielle est intégrée dans le fait, une cellule correspond à un fait détaillé et une couche matricielle correspond à l'ensemble des faits détaillés du domaine spatial couvert par le cube, pour une combinaison de dimensions donnée. Les opérations d'agrégation de mesures correspondant aux faits utilisent des notions d'algèbre de carte (voir section 2.3.1). Une distinction est faite entre un « cube d'objets » et un « cube matriciel » (Figure 32). Un cube d'objets comprend au moins un niveau dimensionnel matriciel mais pas de faits matriciels, et un cube matriciel est entièrement constitué de faits matriciels.

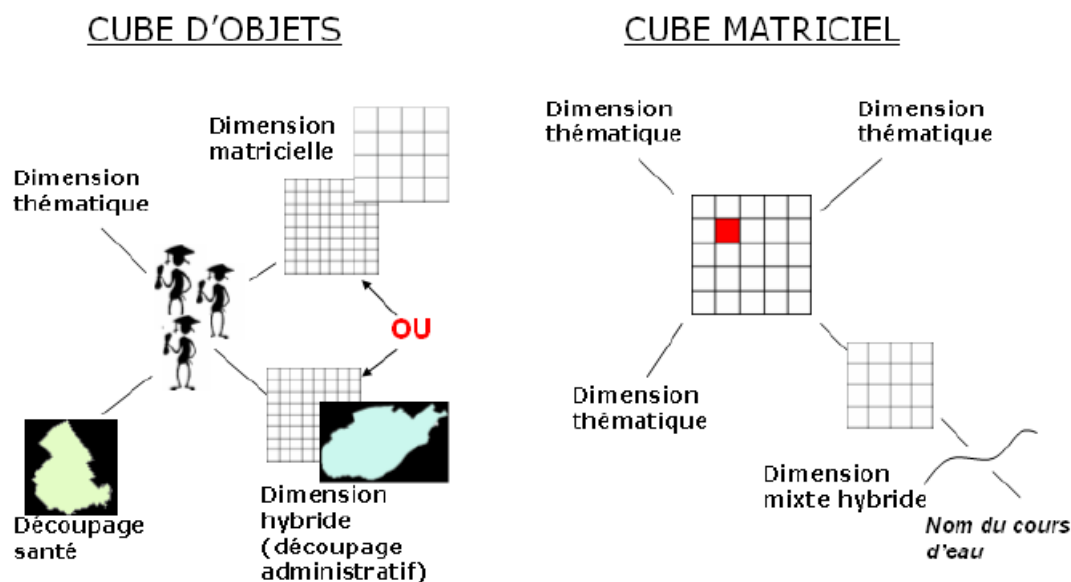


Figure 32. Les deux types de cubes de données intégrant la structure matricielle (Mc Hugh, 2008)

4.1.1.3. Plante (2014)

Plante (2014) reprend les concepts théoriques de McHugh (2008) et démontre le potentiel de la structure matricielle pour l'application d'opérations d'analyse spatiale à la volée dans un SOLAP. Son travail se base sur les cubes matriciels (pas les cubes d'objets) dont il enrichit la définition en proposant plusieurs perspectives d'intégration de la structure matricielle par rapport au fait (Figure 33). Ainsi, un fait détaillé peut être considéré comme une cellule ou comme une couverture matricielle (couverture spatiale⁶¹ modélisée par un raster).

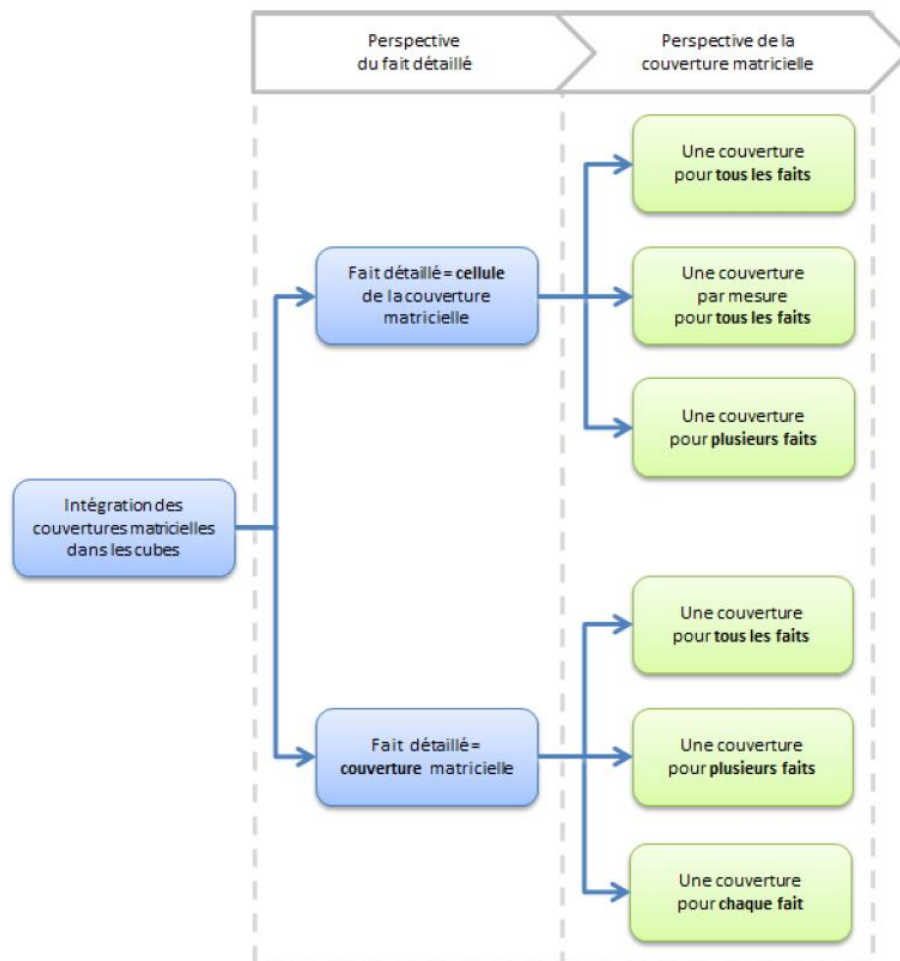


Figure 33. Scénarios d'intégration des couvertures matricielles dans les cubes matriciels (Plante, 2014)

4.1.1.4. Réflexion

Les concepts amenés par les travaux de l'Université Laval constituent une base théorique solide pour l'intégration du raster dans un SOLAP, mais nécessitent néanmoins

⁶¹ Voir section 2.2 pour une définition précise de la notion de couverture en SIG.

quelques précisions par rapport à la structure physique d'un raster telle que définie à la section 2.1.2.2. En effet, les deux types de modèles de Mc Hugh (2008), à savoir cube d'objets et cube matriciel, peuvent être implémentés dans un outil traitant le raster comme du vecteur, c'est-à-dire par séparation des géométries des objets spatiaux (mailles) et de leurs valeurs sémantiques. Cependant, nous avons vu qu'un raster est un tableau et par conséquent, la dissociation des valeurs sémantiques (valeurs du tableau) et géométriques (coordonnées du tableau) implique la disparition de la notion de raster et des performances qui en découlent.

En partant de cette hypothèse, on se rend compte que seuls les cubes matriciels peuvent être implémentés dans un outil SOLAP conservant la structure physique des raster. En effet, les cubes d'objets, où l'information maillée est stockée uniquement par son aspect géométrique, nécessitent une jointure de la dimension matricielle avec les faits pour répondre à une requête spatiale (comme on peut le voir dans un SOLAP vecteur). Dans le cas d'un cube matriciel, la géométrie et les valeurs sémantiques des raster ne font qu'un dans l'ensemble des faits, et les relations avec les dimensions matricielles sont gérées par des opérations d'algèbre de carte entre raster.

Cette constatation ne dénigre pas totalement le modèle par cubes d'objets. En poussant notre réflexion un peu plus loin, on se rend compte que si les faits sont stockés dans un cube MOLAP avec deux dimensions « rangées » et « colonnes » identiques⁶² à celles de la dimension matricielle (une dimension matricielle comprend logiquement deux « sous-dimensions » « rangées » et « colonnes »), les géométries des dimensions matricielles peuvent être mises très efficacement en relation avec les faits par des opérations de type « drill across » entre deux cubes de données⁶³. Ce genre d'opération est d'ailleurs expliqué dans (Gomez *et al*, 2012). De cette manière, la structure physique du raster est partiellement conservée dans le cube MOLAP. Néanmoins, cette approche comporte plusieurs défauts par rapport à un SOLAP conservant totalement la structure physique du raster :

- La structure physique du raster est partiellement conservée uniquement si le SOLAP est basé sur un cube MOLAP. Or, nous avons vu à la section 1.3.5.2 que les outils MOLAP sont presque tous propriétaires et que leur gestion de l'OLAP n'est pas toujours standardisée.
- Les géométries des cellules sont toutes définies selon le mode vectoriel (polygones) alors que cette géométrie est définie implicitement dans la structure physique d'un raster.
- L'intégration de données nativement raster est plus complexe dans un outil nécessitant une restructuration physique de ces derniers. D'un autre côté, nous avons vu que plusieurs SGBD relationnels (PostGIS, Oracle) proposent déjà un modèle pensé pour conserver et exploiter la structure physique des données raster.
- L'intégration à la volée de raster dimensionnels (voir section 5.5.1) dans l'analyse décisionnelle est difficilement envisageable (conséquence du point précédent)

⁶² Par « identique », nous voulons dire que les dimensions partagent les mêmes instances.

⁶³ On considère ici que la dimension matricielle est un cube de données, même si elle ne contient que deux « sous-dimensions » : « rangées » et « colonnes ».

- Les opérations raster doivent être redéfinies dans le serveur OLAP alors qu'elles existent déjà dans les SIG.

L'approche par cubes matricielles semble donc être la bonne stratégie d'intégration du raster au SOLAP puisqu'il s'agit de la seule méthode capable de conserver la notion de raster telle que définie dans les formats de données raster des SIG.

4.1.2. Optimisation des requêtes raster

Même si les modèles SOLAP matriciels n'en sont encore qu'à leurs prémices, certains travaux proposent déjà des méthodes visant l'optimisation des requêtes raster dans les entrepôts de données. En effet, l'optimisation des requêtes, qui est une problématique centrale de l'OLAP en raison de la quantité considérable de données, est d'autant plus importante dans les entrepôts raster car un fait détaillé pris individuellement (selon l'approche par cube matriciel de Mc Hugh) peut potentiellement comporter des millions de pixels, ce qui rend les requêtes multidimensionnelles beaucoup plus complexes.

Gutierrez & Baumann (2008) appliquent le principe de la pré-agrégation des données bien connue des systèmes OLAP (par la matérialisation de cuboïdes) sur une base de données image. Leurs expériences confirment l'intérêt de cette méthode appliquée aux requêtes visant à agréger les données contenues dans des images raster.

Kang *et al* (2013) simplifient les agrégations de raster dans des entrepôts de données relationnels en les regroupant en clusters. Par exemple, plutôt que de sommer les pixels homologues de plusieurs raster (opération locale d'algèbre de carte) appartenant à un même cluster (donc considérés comme similaires), les pixels d'un seul raster sont multipliés par le nombre de raster du cluster concerné par l'opération. Le gain de performance est évident, mais cette technique engendre également une perte de précision. La « clusterisation » est aussi employée dans (Shanmugasundaram *et al*, 1999) pour des cubes non-spatiaux afin de réduire la taille des cubes et d'accélérer les requêtes au détriment de la précision des résultats. Il s'agit d'une alternative intéressante à la matérialisation des cuboïdes qui peut vite s'avérer très lourde.

Notons que ces deux méthodes d'optimisation sont pensées pour une structure de données conservant la notion de raster telle que définie à la section précédente.

4.2. SOLAP continu

La gestion du caractère continu de l'information spatiotemporelle dans un SOLAP a déjà fait l'objet de plusieurs recherches. Cette section a pour but de présenter brièvement les différentes réponses que la littérature propose par rapport à cette problématique. Une comparaison des principales caractéristiques de ces modèles sera ensuite dressée.

Notons que les travaux de Mc Hugh (2008) et Plante (2014) montrent aussi des exemples d'application du modèle « cube matriciel » sur des données spatialement continues. Afin d'éviter les redondances, nous ne repasserons pas sur ces modèles puisqu'ils présentent des caractéristiques (voir Tableau 2) similaires à celui de Gomez *et al* (2010).

4.2.1. Ahmed & Miquel (2005)

Ahmed & Miquel (2005) et Ahmed (2008) étendent le modèle de Vassiliadis⁶⁴ (1998) en y intégrant la notion de dimension continue. Ce nouveau modèle repose sur deux cubes de données : un cube discret (cube interne) et un cube continu (cube externe). Le cube discret stocke les données et est à la base de toutes les opérations SOLAP classiques définies par Vassiliadis (1998). Le cube continu reprend la structure du cube discret en associant des fonctions d'interpolation aux dimensions considérées comme continues (typiquement, l'espace et le temps). Ces dimensions contiennent donc théoriquement une infinité de membres dont les mesures sont interpolées à la volée sur base du cube discret. Ce modèle (Figure 34) offre ainsi une vision continue du cube de données à l'utilisateur, mais permet également de gérer les problèmes de données manquantes ou erronées.

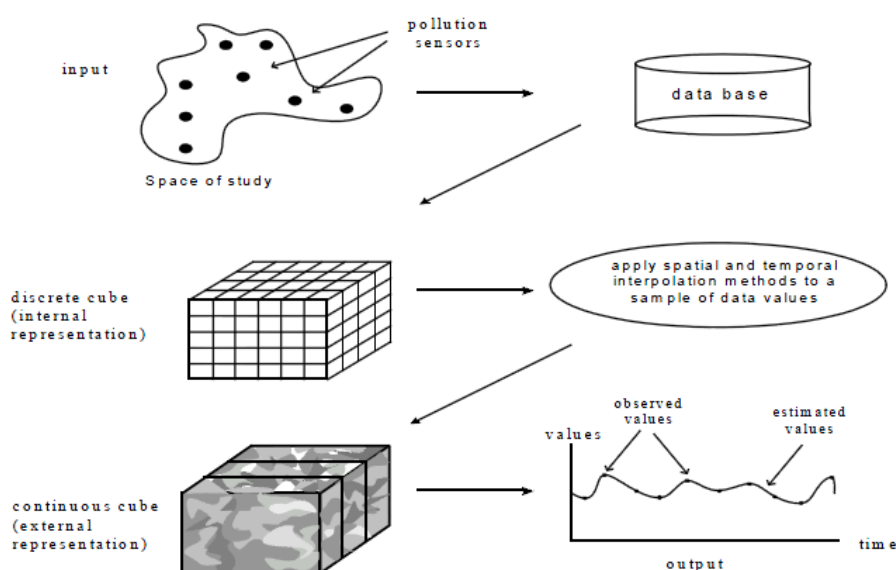


Figure 34. Élaboration d'un cube de données continu (Ahmed & Miquel, 2005)

4.2.2. Vaisman & Zimányi (2009)

Vaisman et Zimányi (2009) intègrent la continuité en définissant un type « champ continu » (ou « field ») pour les dimensions et pour les mesures. Un champ continu peut être uniquement spatial (infinité de valeurs dans un domaine spatial) et/ou temporel (infinité de valeurs dans un domaine temporel). Le formalisme graphique utilisé représente les champs continus sous forme de pictogrammes associés aux attributs des différents éléments de l'entrepôt de données (faits et niveaux de dimensions). Par exemple, dans la Figure 35, l'aptitude des sols (« suitability ») est une mesure spatialement continue, la classification des sols est une dimension spatiale composée de deux niveaux spatialement continus⁶⁵ (les types de sols, eux-mêmes classés en groupes de sols), le ph est une

⁶⁴ Modèle introduisant formellement les notions de dimensions, hiérarchies et cubes ainsi que les opérations sur les cubes à un niveau conceptuel.

⁶⁵ Un phénomène spatialement discret tel que le type de sol peut donc aussi être considéré comme continu dans ce type de modèle.

dimension spatialement continue et la température est une dimension spatio-temporellement continue.

Les dimensions continues ne comportent qu'une seule instance qui comprend une fonction associant une valeur à l'infinité de points couvrant un domaine spatial (ou spatiotemporel) et ne sont donc pas reliés directement aux faits (d'un point de vue relationnel). Une mesure continue (stockée dans les faits) est une fonction associant une valeur à une combinaison de plusieurs facteurs présents dans les dimensions. Ainsi, d'après la Figure 35, l'aptitude des sols (valeur réelle) en un point quelconque est calculée en fonction du type de sol, du ph et de la température (en admettant que l'intersection des domaines spatiaux des différentes dimensions continues ne soit pas nulle). Les mesures « champs continus » peuvent être agrégées en fonction de tous types de dimensions (continues ou non) en utilisant des notions d'algèbre de carte. Ce modèle est défini à un haut niveau d'abstraction, indépendant du type d'implémentation de l'information spatiale (raster ou vecteur).

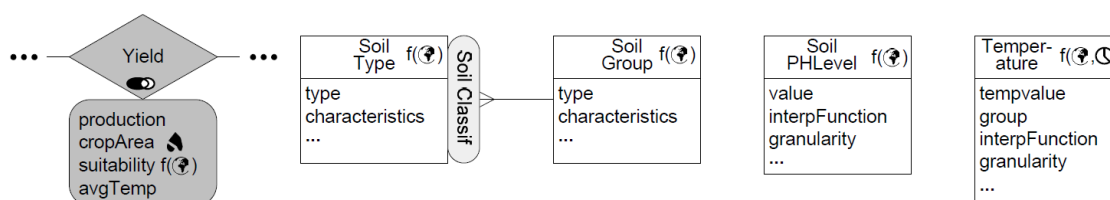


Figure 35. Exemple de dimensions et de mesure de type « champ continu » (Vaisman et Zimányi, 2005)

4.2.3. Gomez et al (2010)

Gomez et al (2010) proposent une implémentation physique du modèle de Vaisman et Zimányi (2009). Cette implémentation est basée sur le SGBD PostGreSQL et son extension spatiale PostGIS⁶⁶. Il s'agit d'un modèle relationnel développé en SQL, et nous avons donc affaire à une architecture ROLAP. Les niveaux de dimensions « champs » (par exemple, la dimension « Soil PHLevel » de la Figure 35) sont des tables contenant un attribut « valeur » et un attribut « géométrie ». Un enregistrement d'une telle table est en réalité une cellule (attribut « géométrie ») provenant d'un raster associée à sa valeur sémantique (attribut « valeur »). Même si les données proviennent d'un raster, la modification de sa structure opérée ici entraîne la destruction du raster au profit d'une structure vectorielle où les cellules sont en réalité des polygones associés à des attributs sémantiques. Ce modèle présente donc des caractéristiques physiques similaires aux modèles proposés par Mc Hugh (2008).

Le champ au niveau de la mesure (par exemple, la mesure « suitability » de la Figure 35) est une fonction prenant les valeurs des dimensions « champs » comme paramètres pour un point dans l'espace. Dans leur modèle, Gomez et al (2010) ne lient pas les dimensions « champ » à la table des faits, si bien que les agrégations des mesures « champ » correspondent à une opération d'algèbre de carte sur base de la fonction stockée dans la mesure « champ », et prenant comme paramètres les dimensions « champ ».

⁶⁶ A cette époque, PostGIS ne gérait pas encore les données raster.

4.2.4. Bimonte & Kang (2010)

Dans Bimonte & Kang (2010) et Bimonte *et al* (2014), il n'y a pas de réelle différence entre membre de dimension et mesure puisque ces deux concepts sont définis par la notion d'objet (Bimonte, 2007). Un objet est un ensemble d'attributs alphanumériques et plus spécifiquement, un objet géographique est un objet comportant au moins un attribut géométrique défini selon le modèle vectoriel. Encore plus spécifiquement, un objet « champ continu » est un objet géographique comportant également un attribut « champ continu ». Ce dernier correspond à un ensemble de fonctions (une par combinaison de dimensions) définies sur des valeurs associées à des points faisant eux-mêmes partie du domaine spatial de l'attribut géométrique.

Puisque mesures et membres sont confondus (et donc permutable dans l'analyse de l'utilisateur), une hiérarchie peut être associée à tous les objets. En particulier, les objets géographiques de type « champ continu » peuvent être associés parallèlement à une hiérarchie définie sur le support « champ » et une autre définie sur un support « non-champ » (qui peut être géométrique ou thématique). Par exemple, la Figure 36 montre une hiérarchie de généralisation (plus on monte de niveau et plus la carte est généralisée) où les géométries sont généralisées au niveau supérieur (Figure 36b : support « géométrie ») et où les points servant de support à la définition du champ continu sont fusionnés au niveau supérieur (Figure 36c : support « champ »). Deux types d'agréations sont donc définis pour l'attribut « champ ». Lorsqu'ils sont agrégés sur le support « non-champ » (dans notre exemple, support « géométrique »), les points définissant le champ sont agrégés par opérations d'algèbre de carte (généralement, une opération locale). Lorsqu'ils sont agrégés sur le support « champ », les points définissant le champ sont agrégés par un opérateur topologique (par exemple, le centre de gravité) au niveau spatial et par une opération d'agrégation classique (somme, moyenne, *etc.*) pour les valeurs sémantiques des points. Après les opérations d'agrégation sur les champs, une valeur en un point quelconque est interpolée à la volée sur base des points qui définissent le champ.

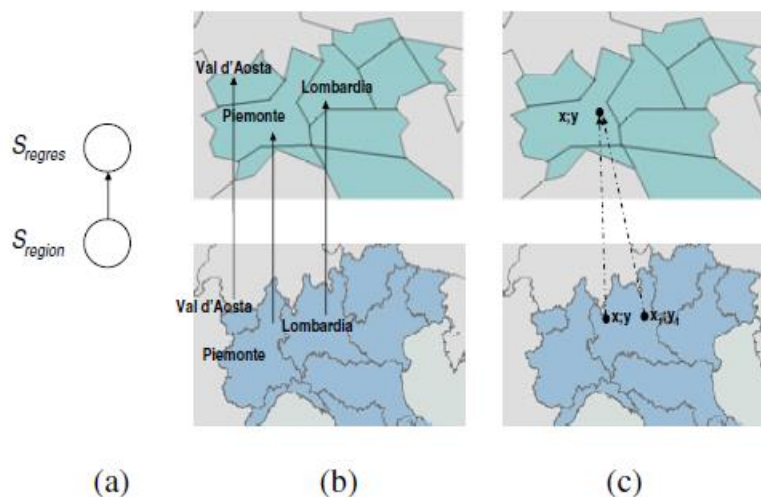


Figure 36. Hiérarchie d'un objet géographique « champ continu ». a) schéma hiérarchique, b) hiérarchie définie sur le support géométrique, c) hiérarchie définie sur le support « champ » (Bimonte & Kang, 2010)

4.2.5. Gomez *et al* (2012)

Gomez *et al* (2012) proposent un autre modèle SOLAP adapté aux champs continus, en étant cette fois complètement indépendant de toute modélisation (physique ou logique) de l'information spatiale. Ainsi, ce modèle peut être potentiellement adapté aux champs modélisés en raster ou en vecteur (polygones de Voronoi, TIN, *etc.*). Dans ce travail, un rapprochement très intéressant est fait entre la notion de champ continu et de cube OLAP. En effet, un champ continu discrétisé (un raster, par exemple) peut être défini comme un cube OLAP et peut donc être comparé à un autre cube OLAP partageant au moins un même niveau de dimension spatiale (avec les mêmes instances) par opérations de type « drill across ». Le modèle est validé par un exemple exploitant des données raster.

4.2.6. Zaamoune *et al* (2013)

Le modèle UML proposé par Zaamoune *et al* (2013) gère les champs continus dans un SOLAP en les définissant comme des dimensions modélisées sous la forme de champs vectoriels (voir section 2.1.1). En particulier, ce modèle supporte les analyses multi-résolutions à travers les différents niveaux de la dimension spatiale « champ incomplet » (définis à plusieurs résolutions spatiales respectives). Dans une première approche, le niveau spatialement continu le plus détaillé est relié à la table de faits et le « roll up » agrège les données du niveau inférieur de manière classique. Dans une seconde approche, la table des faits est reliée à un niveau qui n'est pas nécessairement le niveau le plus détaillé, et le « drill down » interpole à la volée les valeurs d'un niveau supérieur. Puisqu'il s'agit de données vectorielles, la spatialité du modèle est gérée selon une approche ROLAP.

4.2.7. Comparaison

Nous venons de présenter six modèles permettant d'intégrer la continuité spatiale dans le SOLAP. Le Tableau 2 dresse une comparaison de ces modèles (plus ceux de Mc Hugh et Plante) sur base des six paramètres suivants :

- **La méthode d'interpolation des champs** : un champ continu est toujours stocké de manière discrète (voir section 2.1.1). La continuité est introduite par des méthodes d'interpolation qui sont propres à la modélisation spatiale des données.
- **La modélisation spatiale** : certains modèles sont complètement indépendants du type de modélisation spatiale. Les autres utilisent essentiellement le mode vecteur. Le mode raster est clairement sous-représenté⁶⁷ même si la plupart des auteurs s'accordent sur le fait que le raster est une modélisation très répandue pour les phénomènes spatialement continus. Aucun modèle ne met l'accent sur le potentiel de la structure physique du raster.
- **La méthode d'agrégation des champs continus** : la plupart des modèles utilisent l'algèbre de carte de Tomlin pour agréger les champs continus

⁶⁷ Gomez *et al* (2010), Mc Hugh (2008) et Plante (2014) exploitent une structure géométriquement identique au raster (structure maillée), mais s'apparentant plus au vecteur d'un point de vue physique. Le modèle de Gomez *et al* (2012) est le seul à être illustré par un exemple intégrant physiquement le raster.

- **L'emplacement de la continuité spatiale dans la structure multidimensionnelle :** certains modèles considèrent les champs continus uniquement comme des dimensions alors que d'autres les considèrent aussi comme des mesures. Le modèle de Gomez *et al* (2012) est le seul à les considérer comme des cubes OLAP indépendants.
- **Le niveau d'abstraction OLAP:** beaucoup de modèles sont conceptuels et donc décrits de manière indépendante de l'architecture OLAP (MOLAP ou ROLAP). Les modèles physiques utilisent généralement une architecture ROLAP.
- **Permutation des mesures et des dimensions :** le modèle de Bimonte & Kang (2010) est le seul qui autorise ce genre d'opération.

Modèle	Interpolation des champs continus	Modélisation spatiale	Agrégation sur les champs continus	Emplacement de la continuité spatiale dans la structure multidimensionnelle	Niveau d'abstraction OLAP	Permutation des mesures et des dimensions
Ahmed & Miquel (2005)	A la volée	Vecteur (grid)	Classique (sur le cube discret)	Dimension	Conceptuel	Non
Mc Hugh (2008) et Plante (2014)	Implicite (valeur de la maille)	Vecteur (maillé)	Algèbre de carte	Dimension et mesure	Physique (ROLAP ou MOLAP)	Non
Vaisman & Zimányi (2009)	A la volée	Indéfinie	Algèbre de carte	Dimension et mesure	Conceptuel	Non
Gomez <i>et al</i> (2010)	Implicite (valeur de la maille)	Vecteur (maillé)	Algèbre de carte	Dimension et mesure	Physique (ROLAP)	Non
Bimonte & Kang (2010)	A la volée	Vecteur (grid)	Algèbre de carte	Dimension et mesure	Conceptuel	Oui
Gomez <i>et al</i> (2012)	Indéfinie	Indéfinie	Algèbre de carte	Cube à part	Conceptuel	Non
Zaamoune <i>et al</i> (2013)	A la volée	Vecteur (grid)	Classique ou par interpolation	Dimension	Physique (ROLAP)	Non

Tableau 2. Comparaison des différents modèles SOLAP intégrant la continuité spatiale

4.3. Conclusion : hypothèse de recherche

D'après la description des outils SOLAP existants fournie dans le chapitre 3, nous avons décelé trois caractéristiques spatiales importantes qui leur font défaut : l'intégration du mode raster, la possibilité de fournir une analyse continue de l'espace et la gestion de la spatialité directement dans un cube MOLAP.

En analysant la littérature scientifique portant sur la première problématique (l'intégration du raster), nous avons remarqué que les quelques travaux effectués sur le SOLAP raster ouvraient de solides pistes de modélisation. Deux avantages théoriques importants de ce type de modèle sont la gestion du spatialement continu et la possibilité d'appliquer des opérations spatiales à la volée dans l'analyse de l'utilisateur. Cependant, aucun de ces modèles n'a été implémenté en conservant la structure physique du raster (tableau) qui est pourtant un élément clé pour l'optimisation des requêtes spatiales appliquées sur ce modèle de données. Nous avons également constaté que certains travaux portant sur les bases de données raster proposaient des méthodes intéressantes pour optimiser les agrégations raster.

Du côté de l'analyse spatialement continue intégrée au SOLAP, l'état de l'art est plus conséquent et son contenu est plus varié. En effet, depuis une dizaine d'années, des modèles sont proposés à différents niveaux d'abstraction. Certains sont indépendants du type de modélisation de l'information spatiale (vecteur ou raster) alors que d'autres sont valables uniquement en mode vecteur. Le raster, quant à lui, apparaît pratiquement anecdotique dans le sens où il n'a jamais réellement fait l'objet d'une étude exploitant son potentiel physique dans un SOLAP continu. Pourtant, l'algèbre de carte, qui est pensée à la base pour ce type de données, est reprise comme méthode d'agrégation des champs dans la majorité des modèles présentés.

Lors du premier chapitre, une comparaison des trois grands types d'architectures OLAP avait été dressée (ROLAP, MOLAP, HOLAP). Nous y avons constaté que l'architecture MOLAP offrait d'importants avantages pour l'optimisation des requêtes OLAP. Dans le chapitre 2, une description de la structure physique du raster (voir section 2.1.2.2) nous a clairement montré qu'elle s'apparente à un cube MOLAP (voir section 1.3.5.2). La discussion qui a suivi, portant sur les avantages respectifs du mode vecteur et raster, a démontré que le format raster est bien adapté à l'information spatialement continue.

Partant de ces différents constats, notre hypothèse est la suivante : puisque le raster est largement utilisé dans les SIG pour modéliser les champs continus, et que sa structure physique s'apparente à un cube MOLAP, un outil SOLAP considérant le raster comme un cube de données devrait être capable de gérer une analyse multidimensionnelle de l'information spatialement continue, tout en optimisant les agrégations spatiales. Ce genre de solution, qui devrait tirer profit des fonctions raster implémentées dans les outils SIG, permettrait d'intégrer des membres spatiaux à la volée dans l'analyse de l'utilisateur. La composante spatiale des outils SOLAP pourrait ainsi être enrichie afin de renforcer l'intégration de l'OLAP et du SIG.

PARTIE 2 : MODELISATION SOLAP RASTER

« C'est la théorie qui décide de ce que nous pouvons observer. »

Albert Einstein

Chapitre 5 : modèle raster multidimensionnel

5.1. Introduction

Le modèle raster multidimensionnel présenté dans ce chapitre constitue la base du SOLAP présenté dans cette thèse. Il est directement inspiré des architectures MOLAP (voir section 1.3.5.2) et est assimilable à un entrepôt de données physiquement multidimensionnel (cube raster). L'originalité de ce modèle SOLAP est d'exploiter les dimensions spatiales X et Y comme des dimensions MOLAP. Comme dans tous les modèles raster, ces dimensions sont indirectement indexées d'après leurs coordonnées dans l'espace géographique (plan formé par les dimensions X , Y) et les agrégations de mesures dans cet espace sont donc optimisées.

Pour démarrer ce chapitre, nous définirons la notion de « cube raster » en étendant la notion de cube de données physiquement tridimensionnel, ou cube MOLAP (voir section 5.2). Puisque les compléments amenés à la définition d'un cube MOLAP pour en faire un cube raster proviennent de la modélisation raster (voir section 2.1.2.2), un parallélisme sera établi entre le vocabulaire du domaine OLAP et du domaine raster : fait et pixel, membre dimensionnel et coordonnée raster, *etc.* Une attention particulière sera ensuite portée aux dimensions spatiales d'un cube raster (membres et hiérarchies) puis à la troisième dimension du cube (qui peut être spatiale ou non-spatiale) dans les sections 5.3 et 5.4. Il sera ensuite démontré qu'un cube raster présente la caractéristique importante d'accepter l'intégration à la volée d'entités géographiques, appelées membres spatialisés, dans l'analyse de l'utilisateur au moyen d'un raster dimensionnel (section 5.5). La première partie de ce chapitre sera ainsi clôturée par la présentation d'un exemple de cube raster (section 5.6). Cet exemple sera régulièrement utilisé pour illustrer les concepts présentés dans la deuxième partie.

La deuxième partie du chapitre sera consacrée aux opérations permettant la manipulation d'un cube de données raster. Une chaîne d'opérations élémentaires, transparente pour l'utilisateur, sera présentée à la section 5.7. Cette chaîne est essentielle pour calculer les résultats affichés dans les différentes interfaces SOLAP raster présentées à la section 5.8. Une fois tous les concepts de base de notre modèle raster multidimensionnel ainsi définis, nous présenterons et illustrerons les opérations SOLAP orientées utilisateur permettant la navigation dans un cube raster (section 5.9). Lors de la section 5.10, nous discuterons des méthodes d'indexation physique applicables à notre modèle multidimensionnel. Ce chapitre sera naturellement finalisé par une conclusion dans la section 5.11.

5.2. Cube raster

Soit le schéma conceptuel tridimensionnel suivant (Figure 37) :

- Trois ensembles finis, munis d'un ordre total et mutuellement disjoints sont appelés respectivement dimension R (rangée), dimension C (colonne) et dimension B (bande⁶⁸). Un élément d'une dimension est appelé membre.
- Un élément du produit cartésien $R \times C \times B$ est appelé fait détaillé. Pour chaque fait détaillé possible, il existe une mesure atomique appartenant à l'ensemble des réels.
- Un ensemble de faits détaillés est appelé fait. Pour chaque fait, une mesure peut être calculée par agrégation des mesures atomiques des faits détaillés qu'il comprend.

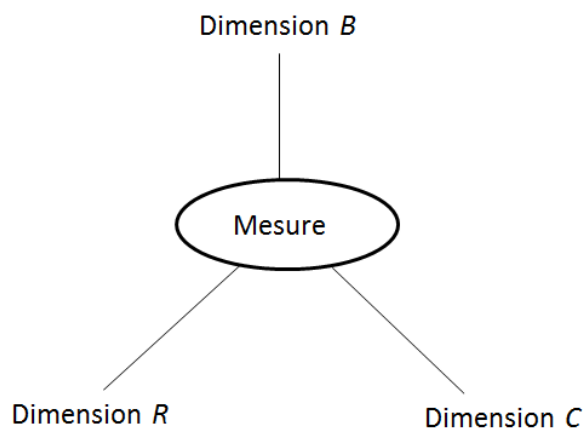


Figure 37. Schéma conceptuel à trois dimensions B , R et C

Ce modèle peut être instancié par un cube de données MOLAP à trois dimensions R , C , B comptant respectivement N_R , N_C , N_B membres ($N_R, N_C, N_B \in \mathbb{N}$) :

$$R = \{0, 1, 2, \dots, N_R - 1\}$$

$$C = \{0, 1, 2, \dots, N_C - 1\}$$

$$B = \{0, 1, 2, \dots, N_B - 1\}$$

Si le nombre total de membres ($N_R * N_C * N_B$) vaut N , un tel cube est une liste ordonnée de N mesures atomiques dont la position physique (ou l'index) est un nombre naturel donné, par exemple⁶⁹, par la formule suivante⁷⁰ ($i, b, r, c \in \mathbb{N}$) :

$$i_{b,r,c} = b * N_R * N_C + r * N_C + c$$

⁶⁸. Le terme « bande » provient de la dimension « bande spectrale » couramment utilisée dans les images satellitaires (raster), et reprise telle quelle dans les SGBD raster. Nous considérons ici la dimension « bande » comme une dimension thématique au sens large, ou éventuellement géométrique si cette dernière exprime la hauteur par rapport à un système de référence altimétrique.

⁶⁹ Indexation classique d'un tableau à 3 dimensions.

⁷⁰ $i_{0,0,0} = 0$ et $i_{N_B-1, N_R-1, N_C-1} = N_B * N_R * N_C - 1 = N - 1$

b , r et c sont les numéros de membres ordonnés selon leur dimension respective B , R et C . Ils forment trois coordonnées (index⁷¹) auxquelles une mesure atomique est strictement associée dans un espace conceptuel discret et tridimensionnel tel que représenté à la Figure 38.

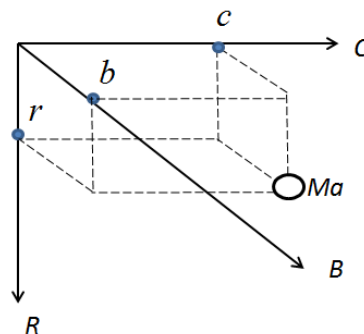


Figure 38. Position d'une mesure atomique en fonction de ses membres de dimension dans l'espace tridimensionnel d'un cube de données

Ce cube de données est un cube raster si les dimensions C et R modélisent respectivement les dimensions spatiales planimétriques X et Y de manière implicite⁷². Pour ce faire, les dimensions C et R doivent être géoréférencées⁷³ grâce à une transformation affine simplifiée⁷⁴ (4 paramètres) permettant la conversion des coordonnées c , r (membres respectifs de C et R) en coordonnées x , y d'un système de coordonnées de référence (SCR) cartographique :

$$x = h_C * c + t_C$$

$$y = h_R * r + t_R$$

avec $x, y, h_C, h_R, t_C, t_R \in \mathbb{R}$

Dans cette formule, x et y sont les coordonnées (ou membres) d'un fait détaillé dans un SCR cartographique (par exemple : Lambert Belge 1972) modélisant l'espace des dimensions X et Y . Les 4 paramètres traduisent mathématiquement 4 sous-transformations entre le plan défini par les dimensions C et R , appelé espace raster, et le plan défini par les dimensions X et Y , appelé espace géographique (Figure 39) :

- Une translation selon C (t_C) et R (t_R)

⁷¹ Trois index logiques (b , r , c) permettent donc de retrouver l'index physique (i) d'un fait détaillé du cube.

⁷² Lorsqu'une dimension est modélisée explicitement dans un cube de données, tous ses membres doivent être sémantiquement définis au moyen de pointeurs (voir section 1.3.5.2).

⁷³ Attention, nous faisons ici référence aux deux systèmes de coordonnées (c , r) et (x , y) implicitement présents dans un raster qui a préalablement été géoréférencé. Il ne s'agit pas du processus de géoréférencement d'une image qui nécessite des transformations plus complexes (sur base de points de contrôle) ainsi que des calculs de rééchantillonnage de la grille numérique raster (valeurs des pixels).

⁷⁴ Valable uniquement pour le cas général où le SCR (X , Y) est aligné avec le système de coordonnées (C , R) de l'espace raster. Les cas plus complexes (et plus rares) doivent être traités avec plus de paramètres (voir section 2.1.2.2).

- Une homothétie selon C (h_C) et R (h_R) qui correspond à l'inverse de la résolution spatiale respectivement en X et en Y (ou la taille d'un pixel en unité de longueur du SCR)

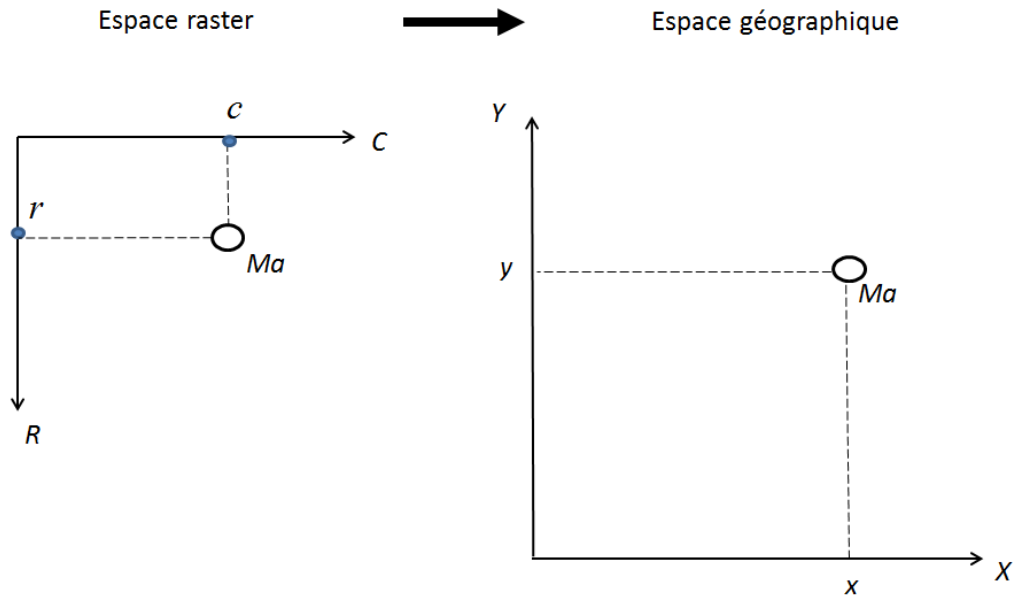


Figure 39. Transformation de l'espace raster vers l'espace géographique

Puisque l'axe R de l'espace raster pointe vers le bas (Sud) et l'axe Y de l'espace géographique pointe vers le haut (Nord), la résolution spatiale en Y est toujours négative. La transformation inverse, de l'espace géographique vers l'espace raster, est logiquement déduite de la transformation directe :

$$c = \frac{x - t_C}{h_C}$$

$$r = \frac{y - t_R}{h_R}$$

En résumé, un cube raster est composé d'une liste ordonnée de mesures atomiques, que nous appelons grille numérique conformément à Collet (1992), et de deux listes de métadonnées : les métadonnées dimensionnelles et les métadonnées spatiales. Jusqu'à présent, les métadonnées spatiales du cube raster sont la seule distinction par rapport à un cube MOLAP⁷⁵ (Figure 40).

⁷⁵ C'est-à-dire un cube de données exploitant les membres dimensionnels comme coordonnées pour indexer les mesures dans des tableaux multidimensionnels (voir section 1.3.5.2).

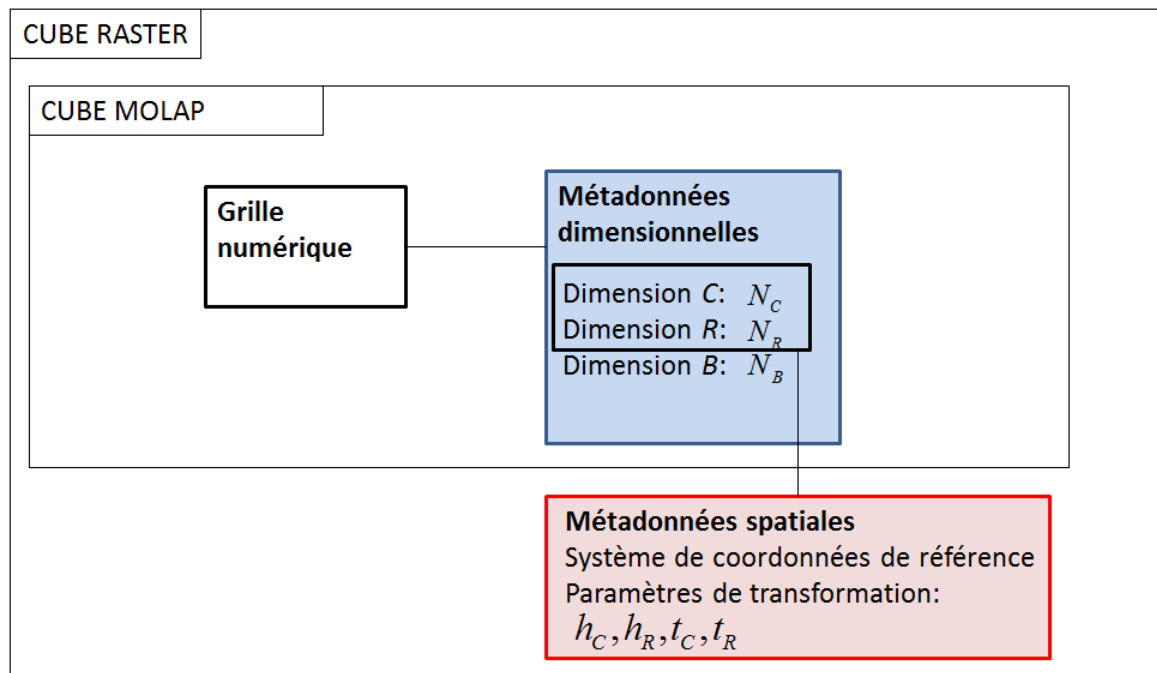


Figure 40. Différence entre un cube raster et un cube MOLAP tridimensionnel

D'après la définition d'un raster dans un SIG, un membre de la dimension R (index r) correspond à la rangée, un membre de la dimension C (index c) correspond à la colonne et un membre de la dimension B (index b) correspond à la bande. Un fait détaillé correspond à un pixel (ou cellule) et une mesure est la valeur d'un pixel que nous notons p . p est une valeur numérique dont le domaine est défini par l'utilisateur.

Notons qu'il serait théoriquement possible de définir un hypercube raster à partir d'un hypercube OLAP comprenant un nombre indéfini de dimensions. Cependant, nous n'avons pas connaissance de raster à plus de trois dimensions. Ajoutons également que les cubes physiques OLAP à plus de trois dimensions peuvent être problématiques pour des raisons d'optimisation et de compression, comme énoncé à la section 1.3.5.2.

5.3. Dimensions raster spatiales (R , C)

Puisque les dimensions C et R (espace raster) d'un cube raster sont transformées vers un système de coordonnées de référence (espace géographique), elles peuvent être considérées comme des dimensions spatiales X et Y , mis à part pour leurs domaines de valeurs respectifs (naturel pour C , R , réel pour X , Y). Le schéma conceptuel tridimensionnel de la Figure 37 peut donc être exprimé comme sur la Figure 41.

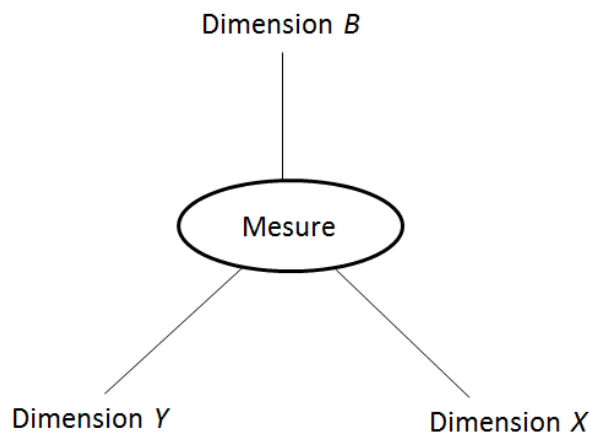


Figure 41. Schéma conceptuel d'un cube raster

Pour étudier le comportement des dimensions spatiales X et Y , il est plus aisé de considérer que le cube raster est uniquement composé de ces dimensions. Ainsi, nous employons toujours le terme « cube raster » lorsque nous faisons intervenir la dimension B . Lorsque nous considérons le cas d'un cube de données avec uniquement deux dimensions spatiales X et Y , nous employons simplement le terme « raster ».

5.3.1. Membres spatiaux

Physiquement, les dimensions spatiales R et C (dont on déduit Y et X) sont chacune composées d'un ensemble de membres permettant l'indexation des mesures dans le cube de données. Conceptuellement, la particularité de ces deux dimensions spatiales nécessite une redéfinition de la notion de membre telle qu'on peut l'observer dans la plupart des systèmes OLAP.

Comme nous l'avons vu à la section 2.1.2.2, une entité spatiale de forme quelconque peut être modélisée par un ensemble de cellules d'un tableau raster. Si nous procédons à la manière d'un cube de données OLAP (et non pas un SOLAP), des entités spatiales quelconques peuvent être représentées par des combinaisons logiques de membres propres à la dimension R et à la dimension C , telles que représentées à la Figure 42.

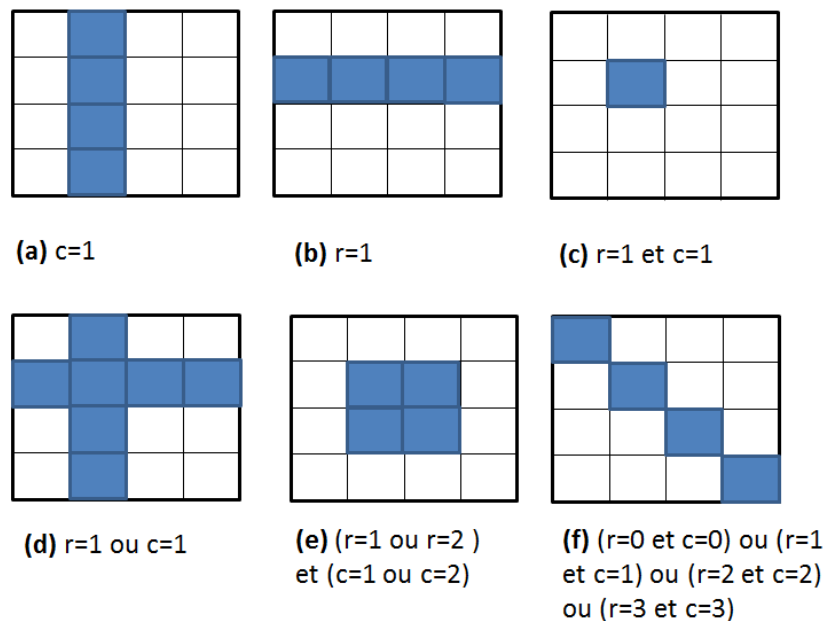


Figure 42. Représentations d'entités spatiales par combinaisons logiques de membres propres à la dimension R et à la dimension C

Cependant, dans tous les outils SOLAP, un membre spatial est toujours défini simultanément sur les deux dimensions de l'espace X et Y car c'est ainsi que sont géométriquement définies les entités vectorielles. Il semble donc logique de procéder de la même manière dans un SOLAP raster en considérant un membre spatial comme un fait impliquant obligatoirement un membre de R (r) et un membre de C (c). Un membre spatial, que nous considérerons toujours comme commun à C et R , a donc un index que nous notons (r,c) . Une mesure d'un fait associé à un membre spatial est notée $p_{r,c}$. Le nombre de membres spatiaux d'un raster est un naturel retrouvé par la formule suivante :

$$N_{R,C} = N_R * N_C$$

Notons finalement que pour les entités spatiales composées d'un seul pixel (Figure 42c) ou représentées par une « diagonale » dans la topologie raster (Figure 42f), la combinaison logique de membres de C et R équivaut à la définition des pixels par leurs coordonnées (comme si les pixels étaient considérés comme des membres communs à C et R).

5.3.2. Hiérarchies spatiales

Jusqu'à présent, nous avons défini quelques propriétés d'un cube raster (géoréférencement et membres communs aux deux dimensions spatiales) en partant de la définition d'un cube MOLAP tridimensionnel. Afin d'introduire progressivement les particularités de notre modèle SOLAP, nous avons émis l'hypothèse que les dimensions du cube de données n'étaient pas hiérarchisées. Cette hypothèse est évidemment très réductrice par rapport aux capacités d'un véritable cube OLAP. En effet, la hiérarchisation des dimensions présente d'importants avantages :

- Une modélisation en harmonie avec la manière dont l'utilisateur se représente les données.
- La possibilité d'intégrer plusieurs niveaux de granularité dans l'analyse.
- Une facilité dans la construction des requêtes, en particulier avec l'emploi du langage MDX. Par exemple, il est possible d'appeler les membres enfants d'une dimension en identifiant uniquement leur membre parent dans une requête du type « drill down ».
- Une optimisation des requêtes due au regroupement physique des membres enfants en relation avec un même parent dans le cube de données⁷⁶. Cette particularité permet une indexation efficace d'une dimension, même lorsqu'elle n'est pas conceptuellement ordonnée⁷⁷.

En réalité, la hiérarchisation des dimensions spatiales d'un cube raster est fondamentalement différente de celle d'un cube OLAP classique en raison de la définition des membres spatiaux énoncée à la section précédente. Puisque les membres spatiaux sont communs aux deux dimensions *X* et *Y*, la hiérarchisation est également commune à ces deux dimensions. La hiérarchisation des dimensions spatiales d'un cube raster peut donc s'avérer assez particulière en partant des concepts de la littérature OLAP. Par contre, le domaine du SIG nous offre une méthode de hiérarchisation simple et couramment utilisée pour les données raster : les pyramides (voir section 2.3.2.2). Cette technique constitue la base de la méthode que nous proposons pour hiérarchiser les dimensions spatiales d'un cube raster.

Nous avons vu qu'une pyramide raster était composée de plusieurs niveaux d'échelle. Chaque niveau comprend un raster, et les raster de chaque niveau couvrent le même domaine spatial (même implantation spatiale) et le même domaine sémantique (les pixels représentent le même phénomène). En revanche, le raster de chaque niveau est caractérisé par une résolution spatiale différente. Plus on monte de niveau, plus la résolution spatiale diminue (donc les nombres de colonnes et de rangées diminuent de la même manière). Nous avons également vu qu'une pyramide raster pouvait être caractérisée par un tuilage à taille⁷⁸ constante sur l'ensemble des niveaux. Dans ce modèle, nous considérons qu'un niveau de pyramide est un raster complet (pas de tuiles). La Figure 43 illustre un exemple de pyramide raster (non tuilée) à trois niveaux caractérisés par un facteur d'échelle de 2 en *X* et en *Y*.

⁷⁶ A condition que le cube de données soit physiquement multidimensionnel, comme dans une architecture MOLAP.

⁷⁷ Exemple de dimension ordonnée : le temps. Exemple de dimension non-ordonnée : les produits d'un magasin.

⁷⁸ $N_R * N_C$

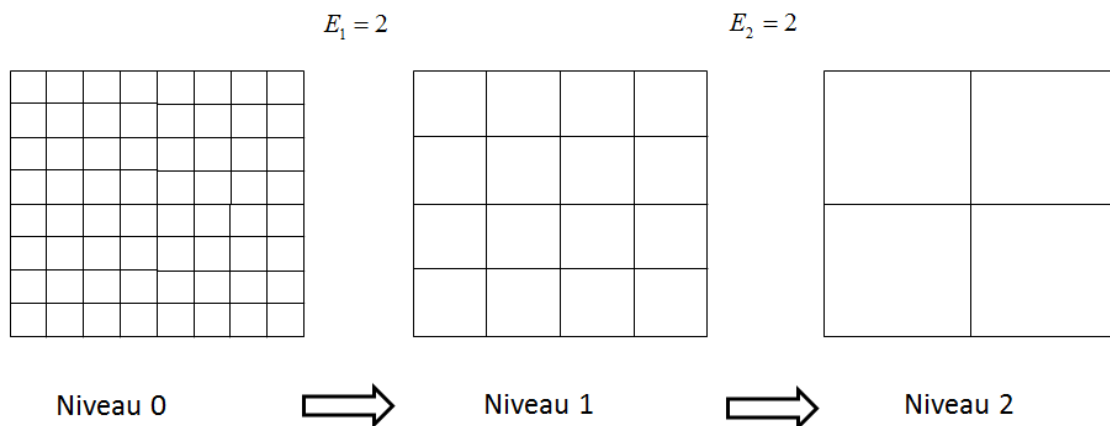


Figure 43. Pyramide raster à 3 niveaux caractérisés par un facteur d'échelle de 2 en X et en Y

La pyramide de la Figure 43 permet la modélisation d'une hiérarchie implicite sur les dimensions spatiales telle que représentée à la Figure 44. Chaque membre spatial enfant est ainsi implicitement relié au membre spatial parent d'un niveau supérieur. Le niveau « all » ne contient qu'un seul membre spatial qui est parent de tous les autres membres. Ce niveau particulier est un raster composé d'un seul pixel. Notons que dans le cas particulier de la Figure 44, la hiérarchie est un tétra-arbre (facteur d'échelle de 2 en X et en Y pour chaque niveau). Si le facteur d'échelle est toujours le même en X et Y , le nombre d'enfants par parent est toujours un multiple de 4. Si le facteur d'échelle entre deux niveaux varie (par exemple, 2 entre les niveaux 0 et 1, et 3 entre les niveaux 1 et 2), le nombre d'enfants par parent est différent d'un niveau à l'autre. Dans le cas très particulier d'un facteur d'échelle différent en X et en Y , le nombre d'enfants par parent est toujours un multiple de 2.

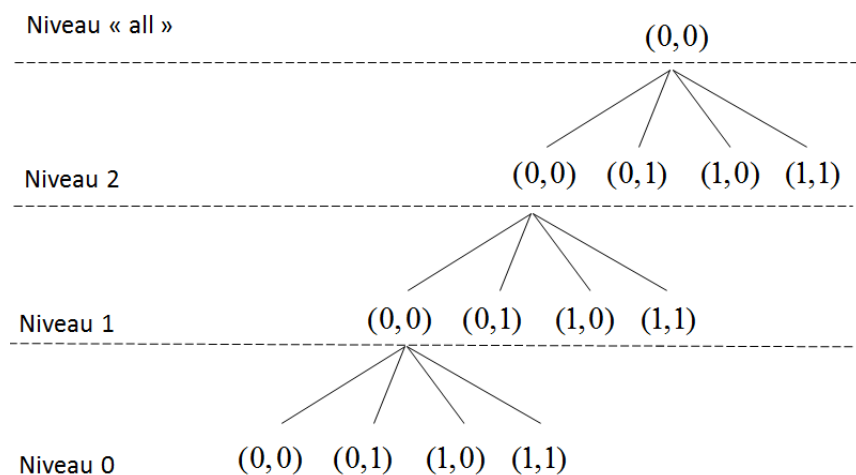


Figure 44. Hiérarchie SOLAP implicite des dimensions spatiales X et Y basée sur une pyramide raster (seuls les enfants du premier membre de chaque niveau sont représentés)

Afin de pas alourdir notre modèle, nous admettons que le facteur d'échelle de chaque niveau est identique en X et en Y . Celui-ci est un nombre naturel non-nul que nous

notons E . E_L est le facteur d'échelle entre un niveau $L-1$ et un niveau L . Le niveau 0 (cube de base) n'a pas de facteur d'échelle, et les différents facteurs d'échelle des autres niveaux sont fixés par l'utilisateur (ce simple paramètre suffit pour définir la hiérarchie spatiale puisqu'elle est implicite). Le nombre de rangées et le nombre de colonnes (ou le nombre de membres R et le nombre de membres C) du raster d'un niveau L sont notés respectivement $N_{R(L)}$ et $N_{C(L)}$. Puisqu'un membre spatial d'un niveau L est une combinaison d'une rangée et d'une colonne du même niveau L , le nombre de membres spatiaux de ce niveau est donné par la formule suivante ($L \in \mathbb{N}$):

$$N_{R(L),C(L)} = N_{R(L)} * N_{C(L)}$$

Le nombre de membres spatiaux d'un niveau L peut aussi être retrouvé à partir du nombre de membres spatiaux du niveau 0 :

$$N_{R(L),C(L)} = \frac{N_{R(0),C(0)}}{\prod_{j=1}^L (E_j)^2}$$

Grâce à cette hiérarchie régulière, les relations entre membres enfants et parents sont implicites dans l'indexation des cellules. Ainsi, la mesure d'un fait associé à un membre spatial d'un niveau L , notée $p_{r(L),c(L)}$, peut être retrouvée par agrégation des mesures des membres enfants du niveau inférieur ($L-1$), en utilisant uniquement les coordonnées des dimensions R et C dans l'espace raster. Si par exemple, la fonction d'agrégation utilisée est la somme⁷⁹, cette mesure peut être calculée par la formule suivante (Figure 45) :

$$p_{r(L),c(L)} = \sum_{j=0}^{E_L-1} \sum_{k=0}^{E_L-1} p_{r(L-1),c(L-1)}$$

$$\text{Avec : } \begin{aligned} r(L-1) &= r(L) * E_L + j \\ c(L-1) &= c(L) * E_L + k \end{aligned}$$

⁷⁹ Notons que cette formule pourrait être exprimée de manière plus générale en utilisant un opérateur binaire « abstrait » associatif et commutatif, puisque la somme est un exemple de fonction d'agrégation OLAP parmi d'autres (moyenne, min, max, etc.). Si cet opérateur est dénoté par $\oplus$, la formule devient :

$p_{r(L),c(L)} = \bigoplus_{j=0}^{E_L-1} \bigoplus_{k=0}^{E_L-1} p_{r(L-1),c(L-1)}$. Cependant, puisque la symbolisation de la somme est supposée connue de tous, il nous semblait plus clair de systématiquement utiliser cet exemple d'agrégation dans notre formalisme mathématique.

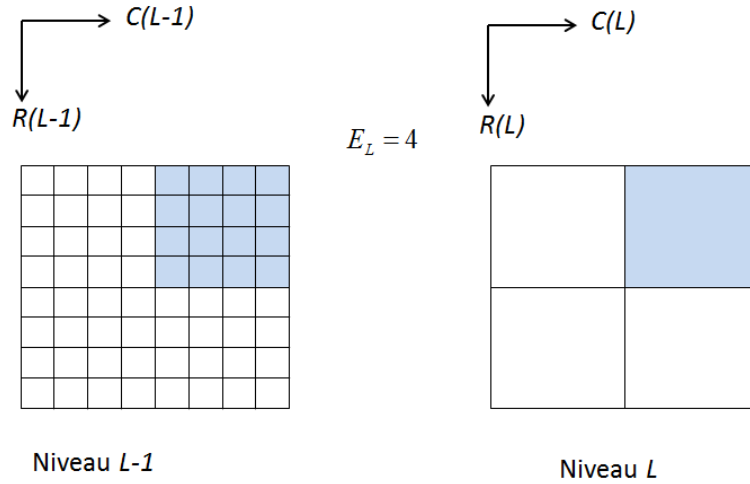


Figure 45. Relation implicite entre membres spatiaux appartenant à deux niveaux hiérarchiques successifs d'un cube raster

Les opérations d'agrégation et de désagrégation en fonction de la hiérarchie des dimensions spatiales sont des forages qui sont nommés respectivement « spatial roll up » (on monte dans la hiérarchie) et « spatial drill down » (on descend dans la hiérarchie). Ceux-ci seront détaillés dans la section 5.9.2. Ce sont les seules opérations SOLAP qui engendrent un changement de résolution spatiale des raster. Elles nécessitent donc une transformation de certaines métadonnées spatiales et dimensionnelles du cube raster.

En ce qui concerne les métadonnées dimensionnelles, seuls les nombre de colonnes et de rangées sont transformés avec une simple division par le facteur d'échelle. Le nombre de membres de la dimension B reste inchangé.

$$N_{C(L)} = \frac{N_{C(L-1)}}{E_L}$$

$$N_{R(L)} = \frac{N_{R(L-1)}}{E_L}$$

$$N_{B(L)} = N_{B(L-1)}$$

En ce qui concerne les métadonnées spatiales, seuls les paramètres d'homothétie sont multipliés par le facteur d'échelle. Les paramètres de translation restent inchangés.

$$h_{C(L)} = h_{C(L-1)} * E_L$$

$$h_{R(L)} = h_{R(L-1)} * E_L$$

$$t_{C(L)} = t_{C(L-1)}$$

$$t_{R(L)} = t_{R(L-1)}$$

La construction d'un niveau de pyramide raster dans un SIG est une opération assez lourde qui est difficilement envisageable à la volée. En particulier, nous avons vu que les opérations OLAP doivent satisfaire des critères de performance qui nécessitent une pré-

agrégation des données en matérialisant certains cuboïdes de données. Puisque dans un SIG, une pyramide raster est toujours pré-calculée pour optimiser les traitements à plusieurs niveaux d'échelle, nous procédons exactement de la même manière dans notre modèle SOLAP. La définition d'une hiérarchie spatiale implique donc systématiquement la matérialisation de cuboïdes raster en fonction de cette hiérarchie (on parle alors de hiérarchie matérialisée). Ainsi, un niveau de hiérarchie spatiale est constitué d'un cuboïde raster pré-calculé à partir du cube raster de base. La Figure 46 donne une représentation d'une hiérarchie spatiale à deux niveaux selon un facteur d'échelle de 2 sur un cube raster tridimensionnel. Remarquons que les métadonnées dimensionnelles de la dimension *B* sont constantes sur les différents niveaux (3 membres dans notre exemple). L'ensemble des cuboïdes matérialisés et du cube de base (niveau le plus détaillé, ou niveau 0) est stocké dans l'entrepôt de données raster.

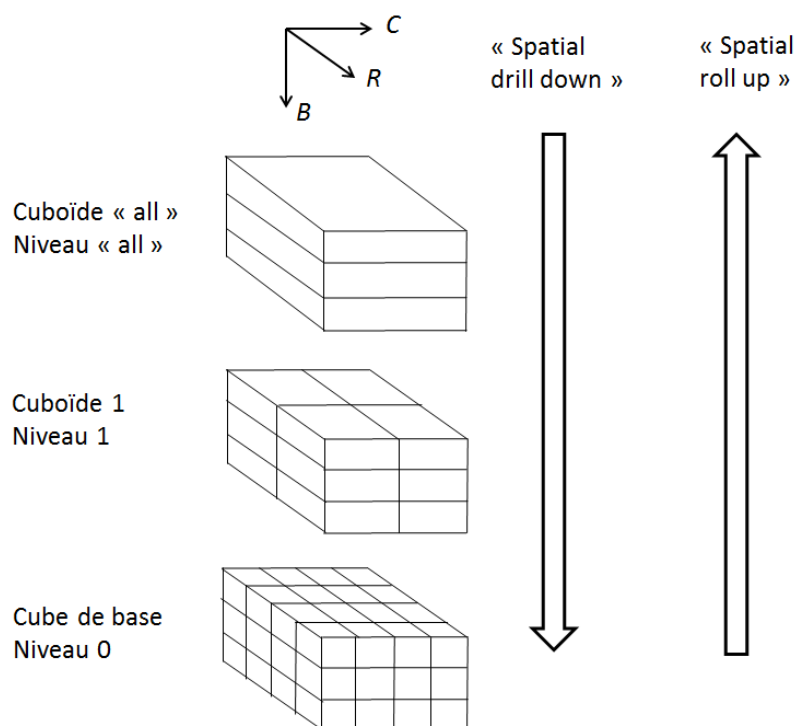


Figure 46. Exemple d'entrepôt de données raster multidimensionnel où la hiérarchie spatiale est matérialisée par des cuboïdes

La hiérarchie des dimensions spatiales a donc deux avantages. D'un côté, les opérations de forage sur cette hiérarchie permettent à l'utilisateur de faire varier la granularité spatiale des données en fonction de différents niveaux d'échelle d'analyse, selon un processus comparable à de la généralisation cartographique. D'un autre côté, la matérialisation de cuboïdes selon les dimensions spatiales (pyramides) pré-calculé les résultats des forages spatiaux (« spatial roll up » et « spatial drill down ») et optimise les traitements à petite échelle (faible résolution) puisque ces derniers sont appliqués sur des cubes raster comportant moins de cellules. Par contre, cette hiérarchisation matérialisée entraîne évidemment une augmentation de l'espace de stockage occupé par l'entrepôt de données. L'espace supplémentaire occupé par un cuboïde issu d'une hiérarchie spatiale est directement lié au nombre de membres contenus dans ce cuboïde (toutes dimensions

confondues). Ce nombre total de membres pour un niveau donné, noté $N(L)$, est donné par la formule suivante :

$$N(L) = \frac{N(0)}{\prod_{j=1}^L (E_j)^2}$$

Finalement, notons que la matérialisation des cuboïdes raster selon la hiérarchie spatiale présente une propriété très intéressante par rapport à un cube de données classique (comportant une hiérarchie par dimension). Nous avons vu à la section 1.2.5 que le nombre de cuboïdes à matérialiser pour optimiser les requêtes OLAP d'un cube de données augmentait de manière exponentielle avec le nombre de dimensions pour lesquelles les opérations de forage sont optimisées (hiérarchie matérialisée). Ainsi, si X et Y présentaient deux hiérarchies distinctes comportant respectivement n_x et n_y niveaux⁸⁰, le nombre de cuboïdes nécessaires pour optimiser les forages sur ces dimensions serait égal à $n_x * n_y$. Grâce à la hiérarchie commune des dimensions spatiales qui caractérise notre modèle, le nombre de cuboïdes nécessaires pour optimiser les forages sur X et Y est simplement égal à $n_{x,y}$ (le nombre de niveaux⁸¹ de la hiérarchie commune à X et Y). La matérialisation de cuboïdes sur ces dimensions est donc une opération particulièrement rentable.

5.4. Troisième Dimension raster (B)

La troisième dimension d'un cube raster est celle que nous appelons dimension B . Dans un cas quelconque (section 5.4.1), ses caractéristiques ne sont pas différentes de celles d'une dimension non-spatiale qu'on retrouve dans un cube de données d'une architecture MOLAP classique. En revanche, lorsque B modélise une dimension conceptuellement continue telle que le temps ou la hauteur (section 5.4.2), elle peut être définie d'une manière pratiquement similaire à celle d'une dimension spatiale (voir section précédente) avec les avantages qui en découlent : hiérarchie et topologie⁸² implicite.

Notons que puisque le modèle tridimensionnel présenté dans ce chapitre sera étendu vers un modèle ROLAP comptant un nombre indéfini de dimensions (voir chapitre 6), B est considéré ici comme une dimension quelconque. Néanmoins, si l'on devait se limiter à cube raster tridimensionnel, la dimension B devrait obligatoirement modéliser le temps puisqu'un entrepôt de données est avant tout un outil d'archivage permettant une analyse selon cette dimension (les deux dimensions R et C étant réservées aux dimensions spatiales Y et X). C'est d'ailleurs le cas dans l'exemple de cube raster présenté à la section 5.6 (analyse spatio-temporelle des cambriolages).

⁸⁰ Niveau « all » compris.

⁸¹ Niveau « all » compris.

⁸² Nous avons vu à la section 2.1.2.2 que lorsque des entités spatiales étaient modélisées par un raster, celles-ci présentaient une topologie implicite, appelée « topologie du jeu d'échec » ou « topologie raster ».

5.4.1. Membres et hiérarchies d'une dimension quelconque

Contrairement aux dimensions spatiales X et Y (ou C et R), une dimension B quelconque se comporte exactement comme dans un cube de données MOLAP classique. Un membre de la dimension B est indexé par sa coordonnée b , et la mesure d'un fait défini sur un membre b de cette dimension est noté p_b . Si une dimension B quelconque est hiérarchisée, cette hiérarchie est explicite et doit donc être clairement définie sur les différentes valeurs de l'index b qui font office de pointeurs vers une série d'attributs caractérisant un membre de B au niveau le plus détaillé. Un niveau de dimension supérieur est alors défini en identifiant les différents membres et leur relation de parenté avec les membres du niveau le plus détaillé. Un troisième niveau de dimension est défini en identifiant les différents membres et leur relation de parenté avec le niveau inférieur, *etc.* La Figure 47 donne un exemple⁸³ de hiérarchisation de la dimension B en utilisant simplement une base de données relationnelle couplée au cube raster (schéma en étoile). A moins que la hiérarchie de la dimension B ne soit complètement régulière (par exemple : tétra-arbre), les relations de parenté entre les membres des différents niveaux ne sont pas implicites comme dans le cas d'une hiérarchisation spatiale.

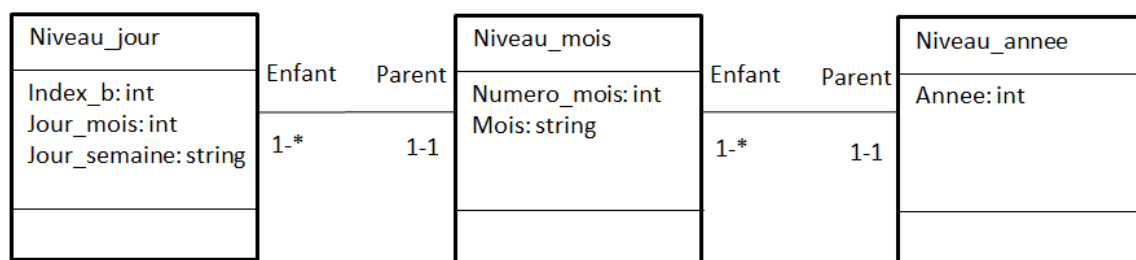


Figure 47. Exemple de hiérarchisation de la dimension B en utilisant une base de données relationnelle (diagramme de classe UML)

Un membre d'un niveau de la dimension B est toujours constitué d'un ensemble de membres du niveau le plus détaillé. Sélectionner un membre d'un niveau⁸⁴ L (par exemple, dans une opération de coupe définie dans la section 5.9.1.1) équivaut donc à sélectionner les membres du niveau le plus détaillé (niveau 0) qui sont hiérarchiquement liés au membre sélectionné. En plus de structurer les données de manière cohérente par rapport à la vision de l'utilisateur, la hiérarchie permet de sémantiquement regrouper les membres détaillés de la dimension B . Ce regroupement peut être exploité dans l'indexation des membres du niveau 0 afin d'optimiser les opérations SOLAP impliquant la dimension B (les membres qui ont un parent commun ont des coordonnées b voisines).

Comme nous l'avons vu à la section 1.2.5, il est également possible de matérialiser des cuboïdes sur la hiérarchie de cette dimension afin d'optimiser les opérations OLAP. Puisque nous considérons que la hiérarchie spatiale est toujours matérialisée (section 5.3.2), un cuboïde devrait ainsi être matérialisé pour chaque élément du produit cartésien

⁸³ Il existe d'autres méthodes pour hiérarchiser une dimension MOLAP. Celle-ci, plutôt utilisée dans le ROLAP, n'est pas forcément la plus efficace dans le MOLAP.

⁸⁴ Attention, L fait cette fois référence à un niveau hiérarchique de la dimension non-spatiale (B).

$H_{X,Y} \times H_B$ où $H_{X,Y}$ est l'ensemble des niveaux de la hiérarchie spatiale et H_B est l'ensemble des niveaux de la hiérarchie de B . Le nombre total de cuboïdes à matérialiser dans l'ED raster pour optimiser les opérations de forage sur toutes les dimensions vaudrait donc $n_{X,Y} * n_B$ ($n_{X,Y}$ est le nombre de niveaux de la hiérarchie spatiale et n_B est le nombre de niveaux de la hiérarchie B , niveau « all » compris). Notons que des pointeurs doivent être également définis sur les niveaux non-détaillés de la hiérarchie de B vers les membres des cuboïdes adéquats.

5.4.2. Membres et hiérarchies d'une dimension continue

Jusqu'à présent, nous nous sommes intéressés au cas général d'une dimension OLAP quelconque stockée dans la dimension B du cube raster. Lorsque B modélise une dimension continue, le cube raster peut être enrichi de propriétés particulièrement intéressantes que nous allons décrire brièvement dans cette section.

Nous avons vu à la section 2.1.3 que le format raster était efficace pour le stockage de l'information spatialement continue car sa structure physique est assimilable à un champ régulier. Ainsi, contrairement aux outils SOLAP vectoriels, notre modèle SOLAP raster est capable d'exploiter la continuité des dimensions spatiales X et Y dans l'indexation des membres spatiaux puisque leur proximité relative dans l'espace géographique est conservée dans l'espace raster (cette propriété sera développée davantage dans le chapitre 7).

Si B stocke une dimension continue de la même manière que R et C , la proximité des entités tridimensionnelles peut également être conservée dans un espace raster 3D. Il suffit alors d'enrichir le cube raster de métadonnées propres à la dimension B similairement aux métadonnées spatiales pour les dimensions R et C . Le cube raster contiendrait ainsi une liste de paramètres supplémentaires :

- Un système de coordonnées de référence (SCR) pour la dimension B
- Un paramètre d'homothétie (h_B) traduisant la longueur d'un pixel dans le SCR de B (ou l'inverse de la résolution selon B)
- Un paramètre de translation (t_B)

Puisqu'une dimension continue est hiérarchisable de manière régulière, les relations de parenté entre les membres des différents niveaux de la hiérarchie de B sont implicites dans leur indexation de manière similaire aux dimensions R et C . Les formules de la section 5.3.2 appliquées aux dimensions R , C sont donc transposables à la dimension B . Ainsi, si E_L est le facteur d'échelle entre un niveau L et un niveau $L-1$ de la hiérarchie d'une dimension B continue, le nombre de membres de B d'un niveau L , noté $N_{B(L)}$, peut être retrouvé à partir du nombre de membres du niveau 0 (niveau de base) :

$$N_{B(L)} = \frac{N_{B(0)}}{\prod_{j=1}^L E_j}$$

Une opération de forage (voir section 5.9.2.1) sur la hiérarchie d'une dimension B continue est résolue grâce aux relations implicites des membres de cette hiérarchie. Par exemple, si la fonction d'agrégation utilisée est la somme, la mesure associée à un fait de coordonnée $B=b$ de niveau L , notée $p_{b(L)}$, est calculée par la formule suivante :

$$p_{b(L)} = \sum_{j=0}^{E-1} p_{b(L-1)}$$

$$\text{Avec : } b(L-1) = b(L) * E_L + j$$

Pour modéliser des phénomènes de la vie courante, il existe quatre⁸⁵ dimensions continues typiques : les trois dimensions spatiales (X, Y, H) et le temps (T). X et Y étant déjà modélisés par les dimensions C et R , H et T sont donc les deux candidats typiques d'une dimension B continue. Ces deux cas particuliers seront détaillés dans les prochaines sections.

Notons qu'il est aussi possible de considérer comme une dimension continue toutes les données numériques que l'on aurait tendance à modéliser intuitivement comme une mesure : un chiffre d'affaire, une température, une pression atmosphérique, *etc.* Par exemple, un cube raster pourrait modéliser une valeur de température (mesure) en fonction de l'espace planimétrique (dimensions X, Y) et de la pression atmosphérique (dimension B continue).

5.4.2.1. Modélisation de la hauteur

Dans le cas où B modélise H (la hauteur), le système de coordonnées de référence est un système altimétrique. h_b correspond à la hauteur d'un pixel⁸⁶ et t_b correspond à la translation de l'espace raster (R, C, B) vers l'espace géographique (X, Y, H) selon B dans le système de référence altimétrique. Les différents niveaux d'une hiérarchie selon B correspondent à des cuboïdes de résolutions altimétriques différentes.

Notons que comme dans notre modèle, X et Y présentent des membres communs, il est également envisageable de considérer que X, Y et H présentent des membres communs (un membre correspond à un volume) et une hiérarchie commune (un niveau correspond à une résolution spatiale 3D). Le nombre de membres d'un niveau hiérarchique commun aux trois dimensions du cube raster serait alors donné par la formule suivante :

$$N(L) = \frac{N(0)}{\prod_{j=1}^L (E_j)^3}$$

⁸⁵ Certaines théories de la physique moderne considèrent que l'univers serait composé de plus de trois dimensions spatiales continues : 10 pour la théorie des super cordes, 26 pour la théorie des cordes bosoniques (Wikipedia, 2015).

⁸⁶ Dans ce cas particulier où un pixel est volumique, on emploie plutôt le terme « voxel » (Lieberwirth, 2008).

Si la fonction d'agrégation est la somme, la mesure d'un fait de niveau L est calculée à partir des mesures des faits du niveau inférieur par la formule suivante :

$$p_{b(L),r(L),c(L)} = \sum_{i=0}^{E_L-1} \sum_{j=0}^{E_L-1} \sum_{k=0}^{E_L-1} p_{b(L-1),r(L-1),c(L-1)}$$

Avec :

$$b(L-1) = b(L) * E_L + i$$

$$r(L-1) = r(L) * E_L + j$$

$$c(L-1) = c(L) * E_L + k$$

Cette modélisation a l'avantage de permettre la matérialisation de la hiérarchie spatiale commune à X, Y, H avec seulement quelques cuboïdes dans l'ED raster (c'est-à-dire un cuboïde par facteur d'échelle commun aux dimensions planimétriques et à la dimension altimétrique). En effet, nous avons vu à la section 1.2.5 qu'avec des dimensions indépendantes (un membre est strictement lié à une dimension), le nombre de cuboïdes à matérialiser augmentait de manière exponentielle avec le nombre de niveaux des dimensions.

5.4.2.2. Modélisation du temps

Dans le cas où B modélise T (le temps), le système de coordonnées de référence est un système temporel. h_B correspond à la durée d'un pixel et t_B correspond à la translation de l'espace raster (R, C, B) vers l'espace géographique (X, Y, T) selon B dans le système de référence temporel. Les différents niveaux d'une hiérarchie selon B correspondent à des cuboïdes de résolutions temporelles différentes.

Notons qu'ici, les relations de parenté entre les membres des différents niveaux hiérarchiques temporels sont implicites uniquement si l'on considère une hiérarchie temporelle régulière. Cela fonctionne donc avec une hiérarchie « seconde-minute-heure-jour-semaine » car il y a toujours le même nombre de secondes par minute, par heure, par jour, et par semaine :

$$E_1 = 60$$

$$E_2 = 60$$

$$E_3 = 24$$

$$E_4 = 7$$

Par contre, cela ne fonctionne pas avec une hiérarchie « jour-mois » car il n'y a pas toujours le même nombre de jours par mois.

5.5. Membres spatialisés (s)

Nous venons de décrire les caractéristiques d'un cube raster par analogie avec un cube MOLAP. Le cube raster est le cœur de notre modèle. Il permet de spatialement représenter des mesures agrégées par cellules de taille constante à différents niveaux de granularité spatiale (pyramides raster) ou thématique (hiérarchie sur la dimension B). Il stocke les faits détaillés et calcule⁸⁷ les faits moins détaillés par agrégation des mesures atomiques en fonction des membres dimensionnels.

Jusqu'à présent, notre modèle ne permet pas de manipuler des entités spatiales modélisées comme des membres dimensionnels clairement identifiables par l'utilisateur d'après leur valeur sémantique. En d'autres termes, la seule définition de membres spatiaux (cellules) ne suffit pas pour un utilisateur qui voudrait manipuler des entités spatiales discrètes telles que des entités administratives (communes, provinces, états, *etc.*), des zones délimitant l'occupation du sol (espace urbanisé, forêt, espace agricole, *etc.*), des routes, des cours d'eau, *etc.* En effet, un membre spatial, tel que défini dans un cube raster, n'a aucune valeur sémantique du point de vue de l'utilisateur. Puisque l'identification de membres spatiaux par leur valeur sémantique est une caractéristique commune à tous les outils SOLAP (voir chapitre 3), notre modèle doit également être capable de gérer cette problématique.

Dans un SOLAP vectoriel, une dimension qui présente à la fois des caractéristiques spatiales (par exemple : implantation géométrique d'une commune) et sémantiques (par exemple : le nom d'une commune) est appelée « dimension géographique » (Bimonte, 2007). Les membres géographiques d'une telle dimension sont généralement définis dans le cube de données au moyen de leur valeur sémantique⁸⁸ et leur spatialisation est possible en les associant à une ou plusieurs géométrie(s) vectorielle(s) via des pointeurs (voir section 3.1.1). Dans la littérature SOLAP, le terme « dimension spatiale » est aussi fréquemment employé pour désigner ce type de dimension. Ce terme présente néanmoins une certaine ambiguïté dans notre modèle puisque nous l'employons déjà pour désigner les dimensions X, Y qui font directement partie de notre schéma en étoile conceptuel (voir Figure 41).

Cette importante différence nous amène à considérer les entités géographiques (définies à la fois sémantiquement et spatialement) de manière très différente par rapport au SOLAP vectoriel. Ces entités, que nous appelons « membres spatialisés », ou encore « membres géographiques », ne sont pas explicitement définies dans notre schéma conceptuel car, en raison de leurs propriétés spatiales et des dimensions X, Y prédéfinies dans le cube raster, celles-ci peuvent être importées à la volée au moment de l'analyse de l'utilisateur et ainsi interagir avec le cube raster⁸⁹. En effet, dans un SOLAP raster, un membre spatialisé s est constitué d'un ensemble de membres spatiaux (cellules) des dimensions X et Y, et cet ensemble correspond à une valeur sémantique (par exemple : la commune de Fléron) que l'utilisateur peut définir librement au moment de son analyse.

⁸⁷ Ou sélectionne des faits pré-calculés dans le cas de la hiérarchie spatiale.

⁸⁸ Une dimension géographique peut donc être associée à une hiérarchie, de la même manière que n'importe quelle dimension OLAP.

⁸⁹ Les entités géographiques (membres spatialisés) ne font donc partie d'aucune dimension prédéfinie dans le cube.

Dans notre modèle raster, un ou plusieurs membres spatialisés peuvent être modélisés par un raster où les valeurs des pixels identifient ces membres de manière purement sémantique (on parle alors de « couche spatialisée » ou « couche géographique »). Ce type de raster, que nous appelons « raster dimensionnel », est très différent du cube raster puisque contrairement à ce dernier, les valeurs des cellules n'expriment pas une mesure mais l'identifiant (pointeur) d'un membre géographique. Un raster dimensionnel est constitué uniquement de deux dimensions R' et C' qui modélisent deux dimensions spatiales Y' et X' . Dans un cas idéal, le raster dimensionnel partage exactement les mêmes métadonnées spatiales et dimensionnelles que le cube raster et par conséquent, $R'=R$ (ou $Y'=Y$) et $C'=C$ (ou $X'=X$). On parle alors de raster dimensionnel intégré. Ce cas particulier sera d'abord décrit dans la section 5.5.1. Dans la section 5.5.2, nous décrirons les conditions nécessaires et suffisantes pour qu'un raster dimensionnel puisse être mis en relation avec un cube raster (cas général). Dans la section 5.5.3, nous décrirons la procédure exacte permettant de transformer un raster dimensionnel quelconque en un raster dimensionnel intégré (intégration à la volée).

5.5.1. Raster dimensionnel intégré

Dans un raster dimensionnel intégré, la valeur s d'une cellule, que nous notons $p'_{r,c}$, est toujours un nombre naturel faisant office de pointeur renvoyant vers une légende. Ainsi, la commune de Fléron aurait par exemple la valeur $s=0$, la commune de Soumagne la valeur $s=1$, etc. Tous les pixels de même valeur s forment un ensemble que nous appelons « membre spatialisé ». Un raster dimensionnel intégré partage exactement les mêmes métadonnées spatiales (h_C, h_R, t_C, t_R et SCR) et dimensionnelles (N_R, N_C) que le cube raster. Ainsi, la relation entre un membre spatialisé et les faits du cube raster est implicite et directe puisque les pixels constituant les membres spatiaux du raster dimensionnel sont indexés de la même manière que les faits du cube raster sur les dimensions spatiales R et C (ils partagent les mêmes coordonnées r et c). La Figure 48 est un exemple d'agrégation d'une mesure associée à un fait dans un cube raster sur base de la sélection d'un membre spatialisé s d'un raster dimensionnel intégré modélisant une couche spatialisée S . Ce genre d'opération est comparable à une opération de type « drill across » dans un cube OLAP.

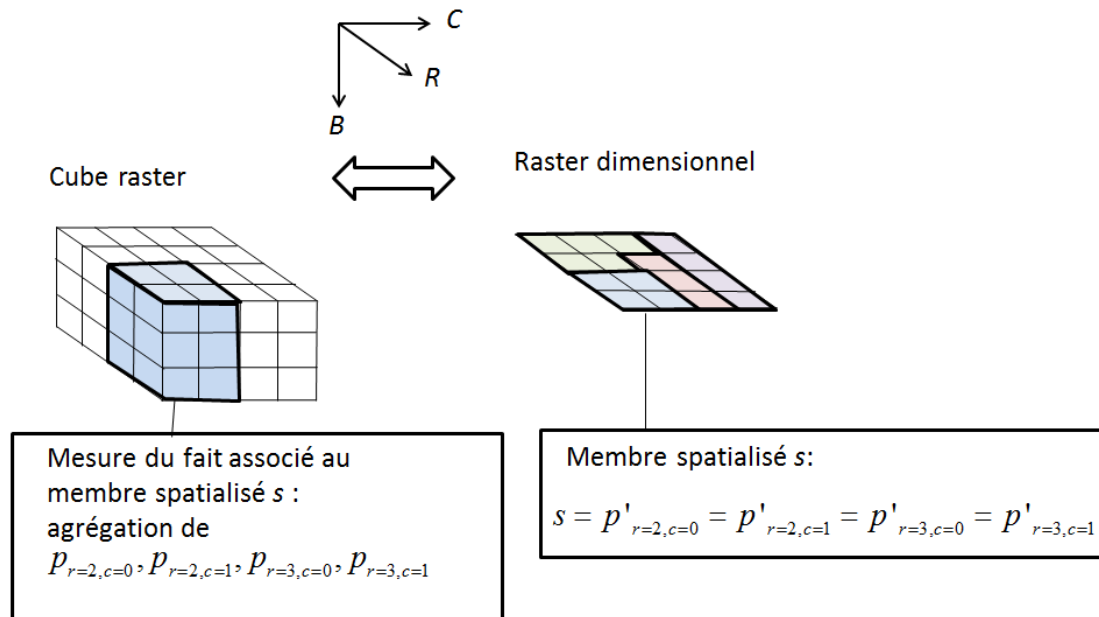


Figure 48. Relation implicite et directe entre un membre spatialisé d'un raster dimensionnel intégré et le fait d'un cube raster

Jusqu'à présent, nos exemples de membres spatialisés portaient sur des phénomènes spatialement discrets, mais il est aussi envisageable de construire un raster dimensionnel à partir d'un champ (voir section 2.1.1). Par exemple, sur base d'un raster de type MNT (modèle numérique de terrain), il est possible de construire un raster dimensionnel en établissant des classes d'altitude. Une classe d'altitude (par exemple : 100 m - 200 m) serait considérée comme un membre spatialisé qui peut être mis en relation avec un cube raster modélisant la température au sol en fonction de l'espace (dimensions R et C) et du temps (dimension B). Cela permettrait de répondre à des questions du type « Quelle est la température moyenne du mois de février 2015 pour les régions spatiales situées à une altitude comprise entre 100 et 200 m? ».

Notons que dans le cas particulier d'un cube raster 3D où B modélise la dimension H (voir section 5.4.2.1), un raster dimensionnel peut contenir une troisième dimension B' (cube raster dimensionnel), afin de mettre des entités spatiales 3D en relation avec le cube raster. Les conditions et méthodes d'intégration sur les métadonnées de B (h_B, t_B , et système de référence altimétrique) sont similaires à celles que nous allons voir dans les prochaines sections pour les métadonnées spatiales (R et C). Ce cas particulier ne sera donc pas détaillé. Il est également possible de modéliser des membres géographiques variant dans l'espace et dans le temps à travers un cube raster dimensionnel où B modélise le temps. Cette perspective intéressante sera discutée dans la section 8.6.3.

5.5.2. Conditions nécessaires et suffisantes sur les métadonnées spatiales d'un raster dimensionnel

L'exemple de la Figure 48 est un cas exceptionnel où les coordonnées r et c sont communes entre le cube raster et le raster dimensionnel (les pixels de mêmes coordonnées r, c partagent la même emprise spatiale). Dans cette section, nous allons étudier les conditions nécessaires et suffisantes sur les métadonnées d'un raster dimensionnel pour

qu'il puisse être mis en relation avec un cube raster. Comme dans toutes opérations de type « drill across » (voir section 1.3.2), le cube raster et le raster dimensionnel doivent partager les mêmes dimensions et les mêmes instances dimensionnelles ($X=X'$ et $Y=Y'$).

En considérant que :

- r et c sont respectivement les coordonnées d'un fait selon les dimensions spatiales R et C du cube raster dans l'espace raster,
- x et y sont respectivement les coordonnées d'un fait selon les dimensions spatiales X et Y du cube raster dans l'espace géographique,
- r' et c' sont respectivement les coordonnées d'un fait selon les dimensions spatiales R' et C' du raster dimensionnel dans l'espace raster,
- x' et y' sont respectivement les coordonnées d'un fait selon les dimensions spatiales X' et Y' du raster dimensionnel dans l'espace géographique.

Pour qu'un raster dimensionnel puisse être mis en relation avec un cube raster, il doit satisfaire les 3 conditions suivantes.

- Le SCR du raster dimensionnel est identique au SCR du cube raster ($X = X'$ et $Y = Y'$).
- La résolution spatiale respectivement en X' et en Y' du raster dimensionnel est identique à la résolution spatiale du cube raster (ou d'un niveau hiérarchique du cube raster) respectivement en X et en Y ($h_C = h_{C'}$ et $h_R = h_{R'}$).
- Si on pose :

$$Z_{R'} = \frac{t_R - t_{R'}}{h_R} \text{ et } Z_{C'} = \frac{t_C - t_{C'}}{h_C},$$

alors $Z_{R'}, Z_{C'} \in \mathbb{Z}$.

La troisième condition est essentielle pour mettre les coordonnées r' et c' du raster dimensionnel en relation avec les coordonnées r et c du cube raster. En effet, nous avons vu à la section 5.2 que le lien entre les dimensions spatiales X, Y et les dimensions spatiales C, R d'un cube raster est donné par les formules suivantes :

$$\begin{aligned} x &= h_C * c + t_C \\ y &= h_R * r + t_R \end{aligned}$$

Puisque les dimensions X, Y et les résolutions spatiales en X, Y sont communes au raster dimensionnel et au cube raster :

$$\begin{aligned} x' &= h_C * c' + t_{C'} \\ y' &= h_R * r' + t_{R'} \end{aligned}$$

Deux membres spatiaux sont communs au raster dimensionnel et au cube raster si

$$x' = x$$

$$y' = y$$

Par conséquent,

$$h_C * c + t_C = h_C * c' + t_{C'}$$

$$h_R * r + t_R = h_R * r' + t_{R'}$$

En isolant c' et r' , on obtient,

$$c' = c + \frac{t_C - t_{C'}}{h_C}$$

$$r' = r + \frac{t_R - t_{R'}}{h_R}$$

En introduisant Z_C et Z_R , on obtient :

$$c' = c + Z_C$$

$$r' = r + Z_R$$

Puisque nous avons vu à la section 5.2 que r (et donc r') et c (et donc c') sont des index dont la valeur doit être un nombre naturel, $Z_{R'}$ et $Z_{C'}$ doivent être des nombres entiers. Ces deux valeurs représentent respectivement les translations sur R et C en nombre entier de pixels (unité de longueur de l'espace raster) entre l'espace raster du cube raster et l'espace raster du raster dimensionnel. La Figure 49 montre ainsi trois exemples de relations entre un cube raster et un raster dimensionnel. Le cas de la Figure 49c est identique à celui de la Figure 48 ($t_R = t_{R'}$ et $t_C = t_{C'}$).

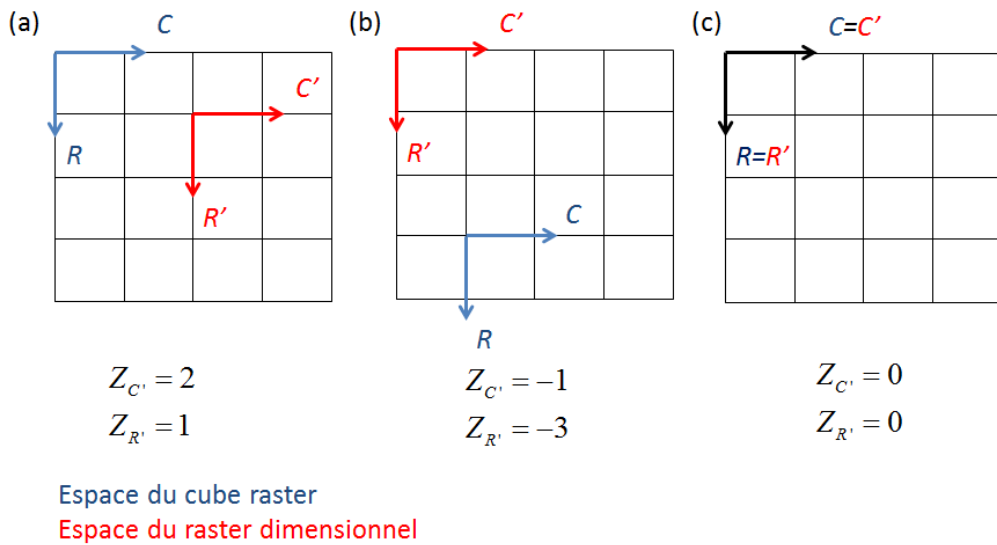


Figure 49. Exemples de relations entre l'espace d'un cube raster et d'un raster dimensionnel

Les translations $Z_{R'}$ et $Z_{C'}$ permettent de retrouver implicitement les relations entre les membres spatiaux du cube raster et les membres spatiaux du raster dimensionnel. Les valeurs de $t_{C'}$ et $t_{R'}$ du raster dimensionnel doivent donc être adaptées pour répondre à la condition $Z_{C'}, Z_{R'} \in \mathbb{Z}$ (voir section 5.5.3.3).

5.5.3. Intégration à la volée d'un raster dimensionnel

Le lien implicite entre un raster dimensionnel et un cube raster par les dimensions spatiales (R et C) est une propriété particulièrement intéressante du SOLAP raster car elle permet l'intégration à la volée de membres spatialisés dans l'analyse de l'utilisateur. Ainsi, un raster dimensionnel peut être construit à partir d'à peu près n'importe quelle source de raster géoréférencé : un fichier, une base de données ou même un autre cube raster.

Afin de satisfaire les trois conditions sur les métadonnées spatiales énoncées à la section précédente, une série de traitements doivent être effectués sur le raster dimensionnel et sur le cube raster au moment de l'intégration. Ces traitements sont expliqués dans les prochaines sections.

Même si le nombre d'étapes d'intégration peut sembler important dans les cas les moins favorables (aucune condition n'est remplie par le raster dimensionnel initial), celles-ci doivent être effectuées une seule fois pour toutes les opérations SOLAP possibles pour un niveau spatial hiérarchique donné (elles n'affectent donc plus les opérations SOLAP impliquant le raster dimensionnel sur ce niveau spatial). Il est également possible d'intégrer un raster dimensionnel pour chaque niveau hiérarchique spatial du cube raster avant l'analyse SOLAP de l'utilisateur (un raster dimensionnel intégré est créé pour chaque niveau spatial).

Notons finalement que, telles que décrites dans prochaines sections, les valeurs de nouvelle grille numérique du raster dimensionnel semblent systématiquement calculées par des méthodes de rééchantillonnage ou d'agrégation à chaque étape d'intégration (cela semblait plus clair de l'expliquer de cette manière). En pratique, la nouvelle grille numérique du raster dimensionnel est calculée en une fois après l'étape 3 (section 5.5.3.3).

5.5.3.1. Transformation du système de coordonnées de référence de l'espace géographique (condition 1)

Puisque le raster dimensionnel doit partager les mêmes dimensions spatiales X et Y (espace géographique) que le cube raster, une nouvelle grille numérique (liste de valeurs des pixels) associée au SCR du cube raster doit être créée pour le raster dimensionnel si celui-ci présente un SCR différent du cube. Puisqu'on passe d'un système X', Y' vers un système X, Y par des méthodes de reprojection cartographique (SCR' vers SCR), les nouvelles valeurs associées à la grille numérique du raster dimensionnel sont rééchantillonnées (Eastman, 2009) grâce à des techniques d'interpolations spécifiques en fonction du domaine sémantique des données (par exemple : interpolation bilinéaire pour du quantitatif, interpolation au plus proche voisin pour du qualitatif).

5.5.3.2. Homothétie de l'espace raster (condition 2)

Une homothétie de l'espace raster du raster dimensionnel est appliquée en fonction d'un niveau L de la hiérarchie spatiale du cube raster. C'est sur ce niveau (cuboïde) que les opérations OLAP intégrant le raster dimensionnel pourront intervenir pour enrichir l'analyse de l'utilisateur (les opérations d'intégration qui suivent doivent donc être répétées sur chaque niveau pour complètement intégrer le raster dimensionnel). La résolution du raster dimensionnel est ensuite adaptée à la résolution du cuboïde raster L ($h_{R'} = h_R$). Une nouvelle grille numérique est créée par une méthode d'agrégation (similaire à celle utilisée dans la construction de pyramides raster) si la résolution doit être diminuée, ou par une méthode d'interpolation (rééchantillonnage similaire à celui cité à la section précédente) si la résolution doit être augmentée. La modification de la résolution spatiale en X et Y du raster dimensionnel correspond à une homothétie de son espace raster respectivement selon C' et R' .

5.5.3.3. Translation de l'espace raster (condition 3)

A partir du cuboïde raster choisi à l'étape précédente, l'espace raster du raster dimensionnel subit une translation selon R' et C' de telle sorte que la troisième condition énoncée à la section 5.5.2 soit remplie ($Z_{R'}, Z_{C'} \in \mathbb{Z}$). Ainsi, si $Z_{R'}$ et $Z_{C'}$ sont arrondis à l'unité la plus proche, $\delta t_{R'}$ et $\delta t_{C'}$ sont respectivement les valeurs réelles à ajouter à $t_{R'}$ et $t_{C'}$ pour respecter la condition de translation de l'espace raster :

$$Z_{R'} = \frac{t_R - (t_{R'} + \delta t_{R'})}{h_R}$$
$$Z_{C'} = \frac{t_C - (t_{C'} + \delta t_{C'})}{h_C}$$

En isolant $\delta t_{R'}$ et $\delta t_{C'}$, on obtient :

$$\delta t_{R'} = t_R - t_{R'} - h_R * Z_{R'}$$
$$\delta t_{C'} = t_C - t_{C'} - h_C * Z_{C'}$$

Une nouvelle grille numérique est créée par rééchantillonnage pour le raster dimensionnel dont les paramètres de translation de l'espace raster ont été modifiés.

5.5.3.4. Optimisation par découpage

Une fois les trois premières étapes effectuées, toutes les conditions nécessaires et suffisantes sont remplies pour qu'un raster dimensionnel puisse être mis en relation avec un cuboïde raster de la hiérarchie spatiale de l'entrepôt raster. Cependant, sauf dans un cas fortuit, le cube raster et le raster dimensionnel ne partagent pas exactement le même domaine spatial. Par conséquent, plusieurs membres spatiaux du cuboïde d'un niveau spatial ne sont pas reliés aux membres spatiaux du raster dimensionnel et *vice versa*. Les opérations OLAP sur des cubes physiques étant effectuées par balayage de toutes les mesures atomiques le long des dimensions concernées, celles-ci sont inutilement alourdies

par la présence de membres spatiaux situés en dehors du domaine spatial commun au cuboïde et au raster dimensionnel. Afin d'optimiser les traitements, un découpage du raster dimensionnel et/ou du cuboïde raster L est effectué pour ne garder que les membres faisant partie du domaine spatial commun. Cette opération est illustrée dans la Figure 50.

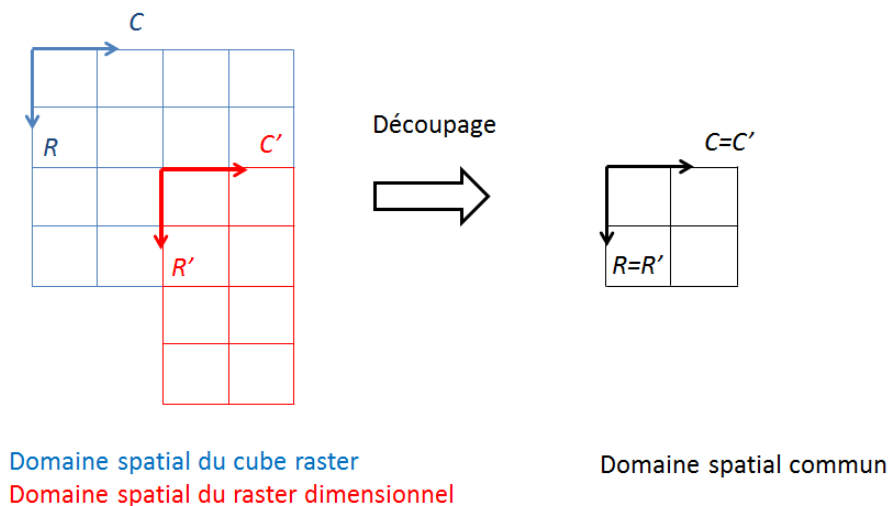


Figure 50. Découpage d'un cube raster et d'un raster dimensionnel pour exclure les membres spatiaux non-communs

Lorsque qu'un raster est découpé (ou filtré), on parle de sous-raster et lorsque qu'un cube (ou un cuboïde) raster est découpé (ou filtré), on parle de sous-cube (ou sous-cuboïde) raster. Il est essentiel de ne pas confondre la notion de cuboïde avec la notion de sous-cube. Dans un cuboïde, les mesures sont indexées par des membres de dimension appartenant à un niveau hiérarchique qui n'est pas le niveau le plus détaillé. Dans un sous-cube, les mesures sont indexées par des membres du même niveau que le cube découpé (niveau le plus détaillé s'il s'agit du cube de base). Le nombre de membres est simplement réduit sur la dimension découpée.

D'un point de vue OLAP, cette opération de découpage correspond à une double opération de coupe (réduction du nombre de membres d'une dimension) sur les dimensions R et C du cuboïde et/ou sur les dimensions R' et C' du raster dimensionnel. Soient N_R , N_C le nombre de membres des dimensions R et C du cuboïde raster, $N_{R'}$, $N_{C'}$ le nombre de membre des dimensions R' et C' du raster dimensionnel, un sous-cube (ou sous-cuboïde) raster et/ou un sous-raster sont calculés à partir du cuboïde raster de niveau L et du raster dimensionnel en suivant cet algorithme de découpage⁹⁰ :

⁹⁰ On admet qu'une suppression des membres d'une dimension entraîne une modification du paramètre N de cette dimension en conséquence.

- Si $Z_{R'} > 0$:
 - o suppression des membres R de coordonnée $r \in [0, Z_{R'}[$
 - o $t_R = t_{R'}$
- Si $Z_{R'} < 0$:
 - o suppression des membres R' de coordonnée $r' \in [0, Z_{R'}[$
 - o $t_{R'} = t_R$
- Si $N_R > N_{R'}$: suppression des membres R de coordonnée $r \in [N_{R'}, N_R[$
- Si $N_R < N_{R'}$: suppression des membres R' de coordonnée $r' \in [N_R, N_{R'}[$
- Si $Z_{C'} > 0$:
 - o suppression des membres C de coordonnée $c \in [0, Z_{C'}[$
 - o $t_C = t_{C'}$
- Si $Z_{C'} < 0$:
 - o suppression des membres C' de coordonnée $c' \in [0, Z_{C'}[$
 - o $t_{C'} = t_C$
- Si $N_C > N_{C'}$: suppression des membres C de coordonnée $c \in [N_{C'}, N_C[$
- Si $N_C < N_{C'}$: suppression des membres C' de coordonnée $c' \in [N_C, N_{C'}[$

Le sous-cuboïde et le sous-raster dimensionnel résultants partagent ainsi exactement les mêmes métadonnées spatiales et les mêmes métadonnées dimensionnelles. On se retrouve donc dans la configuration idéale où les relations entre R et R' et entre C et C' sont implicites et directes ($R=R'$ et $C=C'$). Après cette opération de découpage, le raster dimensionnel est appelé « raster dimensionnel intégré ».

5.5.4. Intégration à la volée d'une couche vectorielle

Nous avons vu qu'un raster dimensionnel pouvait être intégré à un cube raster à partir de n'importe quelle source raster. N'importe quelle source vectorielle peut également être à la base d'un raster dimensionnel en suivant un processus de rasterisation. Lorsque la couche vectorielle à intégrer présente le même SCR que le cube raster, pour un niveau de hiérarchie spatiale L , la rasterisation utilise comme paramètres les métadonnées dimensionnelles et les métadonnées spatiales du cuboïde L . Pour plus de détails, nous renvoyons vers la littérature sur le sujet (Eastman, 2009). Au terme de ce traitement, on obtient un raster dimensionnel intégré partageant les mêmes métadonnées spatiales et les mêmes métadonnées dimensionnelles sur R et C que le cuboïde raster de niveau L .

5.6. Exemple de cube raster

Plusieurs exemples dans les prochaines sections seront basés sur le même schéma de cube raster que celui que nous présentons ici. Ce schéma exploite des données qui concernent des cambriolages situés à Londres pour l'année 2012. Elles sont fournies par la « City of London Police » et la « Metropolitan Police » (UK Police, 2014).

Le modèle conceptuel est représenté par un schéma en étoile à trois dimensions (Figure 51). La mesure est le nombre de cambriolages. Les dimensions spatiales X et Y

sont hiérarchisées selon deux niveaux en fonction de leur résolution spatiale⁹¹ : « pixel = 1000 m » et « pixel = 300 m » (niveau détaillé). La dimension B modélise le temps et est hiérarchisée selon deux niveaux : les trimestres et les mois (niveau détaillé). Pour certaines opérations, nous utiliserons une couche de membres spatialisés modélisés par un raster dimensionnel : les zones représentant l'emprise spatiale des différentes forces de police sur le territoire. Notons que nous n'avons pas mentionné cette « dimension » dans le schéma de la Figure 51 puisque celle-ci est intégrée à la volée au moment de l'opération SOLAP qui la concerne. Dans cet exemple, les mesures sont toujours agrégées par la somme.

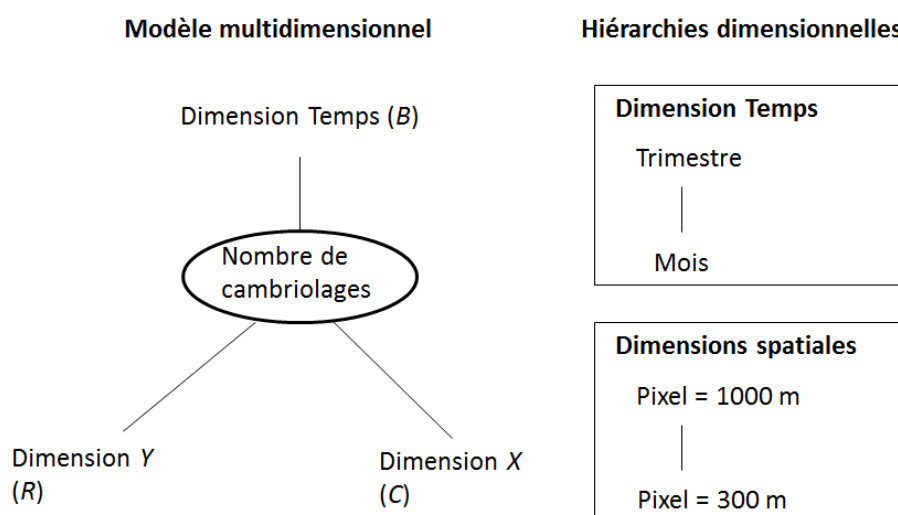


Figure 51. Exemple de cube raster (schéma en étoile)

Remarquons que ce schéma est en contradiction avec le développement théorique de la section 5.3.2 sur les hiérarchies spatiales. En effet, nous avons démontré que, pour que les membres spatiaux de niveaux hiérarchiques différents puissent être mis en relation, le facteur d'échelle entre deux niveaux (E_L) devait être un nombre naturel. Or, dans ce schéma, le facteur d'échelle entre le niveau « pixel = 300 m » et le niveau « pixel = 1000 m » est un réel (3,3333...). Lorsque le facteur d'échelle n'est pas un nombre naturel, les pixels ne sont pas parfaitement alignés entre deux niveaux spatiaux et par conséquent, la hiérarchie spatiale est non-stricte (voir 1.2.1) puisqu'un membre enfant peut avoir plusieurs parents. Pour retrouver la valeur d'un membre spatial parent par rapport à celle de ses enfants, les enfants comportant plusieurs parents devraient être traités de manière spécifique, par exemple par agrégation pondérée en fonction de leur surface d'intersection avec le membre parent traité⁹². La complexité des calculs de forage spatial amené par ces traitements les rend difficilement envisageables au moment de l'analyse. Néanmoins, puisque notre modèle est basé sur des niveaux spatiaux pré-calculés (cuboïdes matérialisés), tous les forages spatiaux sont également pré-calculés (voir section 5.9.2) et les facteurs d'échelles peuvent donc appartenir au domaine des réels strictement supérieurs à un.

⁹¹ Afin de ne pas être trop exhaustif, lorsque nous utilisons la notation « pixel = x m », x est toujours à la fois la dimension d'un pixel en X et en -Y (car l'axe Y pointe vers le bas). Nous n'illustrons pas de cas où la résolution spatiale d'un niveau est différente en X et en -Y.

⁹² Une telle méthode est présentée par Bimonte (2007).

5.7. Opérations élémentaires

Dans cette section, nous présentons une⁹³ chaîne de traitements permettant de calculer le résultat affiché par une interface SOLAP à partir de l'entrepôt de données raster multidimensionnel (qui contient le cube de base et tous les cuboïdes matérialisés). Les mailles de cette chaîne sont appelées « opérations élémentaires ». Elles sont divisées en deux grandes catégories : les filtrages et les agrégations. De manière générale, les filtrages doivent toujours être appliqués avant les agrégations afin de simplifier ces dernières. Les différentes opérations élémentaires sont donc appliquées dans un ordre précis :

- Filtrage vertical
- Filtrage horizontal
- Agrégation partielle
- Agrégation totale

Le résultat produit par une de ces opérations élémentaires est toujours un cube raster⁹⁴ dont les métadonnées ont été modifiées en conséquence. Puisqu'il s'agit d'une chaîne, toutes ces opérations sont elles-mêmes appliquées sur un cube raster⁹⁵, à l'exception de la première opération (filtrage vertical) qui est appliquée sur l'entrepôt de données raster. A chaque étape de la chaîne d'opérations élémentaires, la dimensionnalité du cube raster sélectionné dans la première opération est de plus en plus réduite (réduction du nombre de membres et de dimensions), jusqu'à obtenir la forme souhaitée par l'interface utilisateur (raster ou tableau à une ou deux dimensions).

Dans les prochaines sections, nous présentons les différents types de filtrages et d'agrégations dans l'ordre de leur apparition dans la chaîne de traitements. Cette chaîne est schématisée dans la Figure 52. Remarquons que certaines de ces opérations sont divisées en opérations spatiales (appliquées sur les dimensions spatiales ou sur les membres spatialisés) et non-spatiales (appliquées sur les dimensions non-spatiales). Cette distinction sera également détaillée dans les prochaines sections. Les interfaces se trouvant au bout de la chaîne seront présentées dans la section 5.8.

⁹³ Il s'agit bien d'un exemple. Comme nous le verrons, il est possible d'appliquer certaines opérations élémentaires dans un autre ordre relatif que celui proposé par la Figure 52.

⁹⁴ qui peut prendre la forme d'un cuboïde ou d'un sous-cube.

⁹⁵ qui peut prendre la forme d'un cuboïde ou d'un sous-cube.

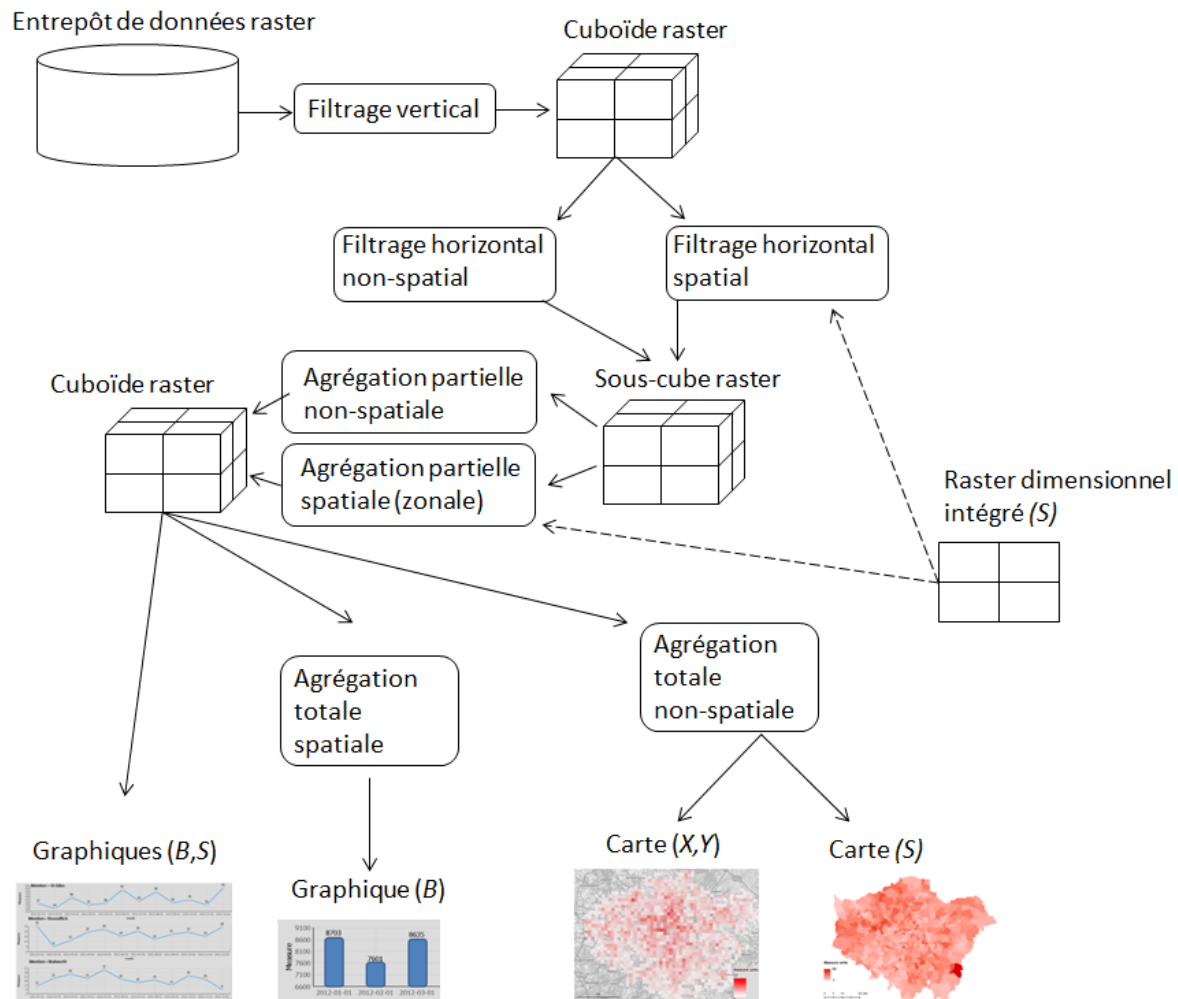


Figure 52. Chaîne d'opérations élémentaires appliquée sur un entrepôt de données raster multidimensionnel

5.7.1. Filtrage vertical

Le filtrage vertical est la première opération de la chaîne qui est directement appliquée sur l'entrepôt de données raster. Ce dernier contient tous les cuboïdes pré-calculés des dimensions dont la hiérarchie est matérialisée, y compris le cube de base. Le filtrage vertical est une sélection des niveaux sur ces hiérarchies et équivaut donc à sélectionner le bon cuboïde dans l'entrepôt de données raster. Les opérations élémentaires suivantes (filtrage horizontal) sont appliquées sur le cuboïde résultant du filtrage vertical.

Puisque nous considérons que la hiérarchie des dimensions X , Y est toujours matérialisée (pyramides raster), le filtrage vertical est toujours au moins spatial : un niveau hiérarchique des dimensions X , Y est sélectionné (par exemple, niveau « pixel = 1000 m »). Éventuellement, la hiérarchie de la dimension B est aussi matérialisée. Dans ce cas, le filtrage est à la fois spatial et non spatial car le cuboïde sélectionné correspond à la fois à un niveau hiérarchique de X , Y (par exemple : niveau « pixel = 1000 ») et à un niveau hiérarchique de B (par exemple : niveau « trimestre » d'une dimension temporelle). Dans le cas particulier où B modélise la dimension H de manière continue et que la hiérarchie de

H est matérialisée, le filtrage vertical est un filtrage spatial 3D (par exemple : niveau « pixel = 1000 m » selon X, Y, H).

La modification des paramètres de filtrage vertical, à savoir le niveau choisi pour les dimensions dont la hiérarchie est matérialisée, équivaut à une opération SOLAP de forage sur ces dimensions (voir section 5.9.2). Si pour toutes ces dimensions, le niveau hiérarchique choisi est le plus détaillé, le cuboïde résultant est le cube de base. Un exemple de filtrage vertical et spatial (X, Y) était illustré dans la Figure 46.

5.7.2. Filtrage horizontal

Les opérations de filtrage horizontal sélectionnent toutes les mesures d'un cube raster en relation avec un ou plusieurs membre(s) (filtre) d'une dimension. Le résultat est un sous-cube raster où le nombre de membres de la dimension filtrée est réduit au nombre de filtres et où l'ordre des membres les uns par rapport aux autres est conservé (les opérations élémentaires suivantes sont donc appliquées sur le sous-cube résultant des filtres horizontaux). Les filtres d'un même filtrage horizontal sont combinés par des opérations logiques « ou » et plusieurs filtres horizontaux sont combinés par des opérations logiques « et ». Comme nous le verrons à la section 5.9.1, le filtrage horizontal est la base de l'opération SOLAP de coupe.

Nous faisons la distinction entre un filtrage horizontal non-spatial (appliqué sur B) et un filtrage horizontal spatial (appliqué sur R et C). Cette différence est illustrée dans la Figure 53 et est expliquée dans les prochaines sections. Nous verrons à la section 5.7.2.2 que physiquement, appliquer un filtrage spatial sur les dimensions R, C ou appliquer un filtrage sur des membres spatialisés s revient exactement au même.

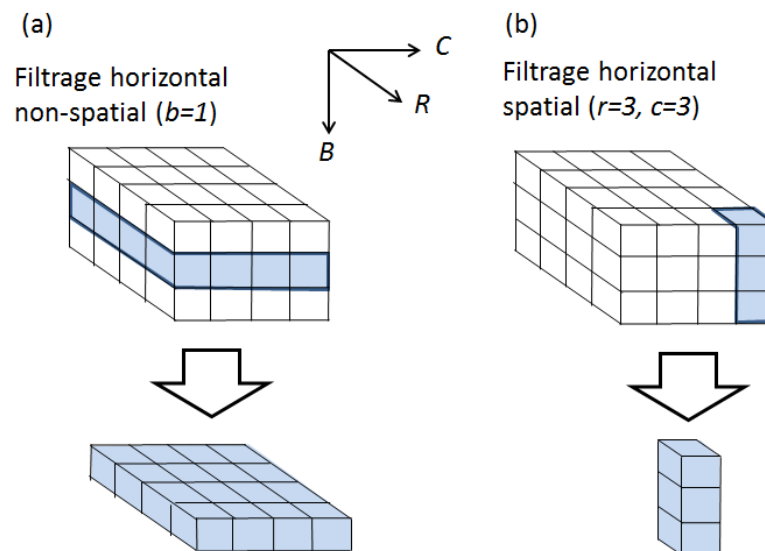


Figure 53. Les deux types de filtres horizontaux d'un cube raster

5.7.2.1. Filtrage horizontal non-spatial

Le résultat d'un filtrage horizontal sur un membre non-spatial (dimension B) de coordonnée b sur un cube raster est un sous-cube comprenant tous les pixels du cube de coordonnée $B=b$ dans le même ordre relatif que celui du cube. Un exemple de filtrage non-spatial à un filtre est donné dans la Figure 53a. Dans un sous-cube résultant d'un filtrage non-spatial à f filtres⁹⁶,

$$N_B = f .$$

La réduction du nombre de membres de B dans le sous-cube résultant entraîne une réorganisation des pointeurs des membres détaillés de la hiérarchie de B si cette hiérarchie est explicite (dimension quelconque). Dans cette réorganisation des membres détaillés, l'ordre relatif des filtres les uns par rapport aux autres est conservé. Un exemple est donné dans la Figure 54.

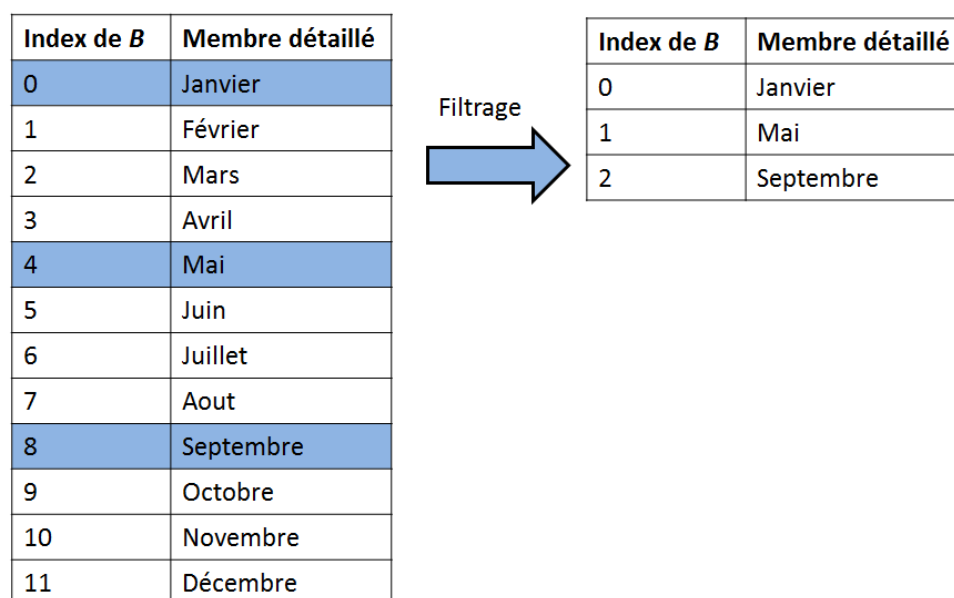


Figure 54. Redéfinition des pointeurs du niveau détaillé d'une dimension B après une opération de filtrage

Si la hiérarchie de B est implicite (dimension continue), les valeurs d'index de B dans l'espace raster sont mises en relation avec le niveau le plus détaillé de leur espace conceptuel (par exemple : le nombre de mois) par simple modification des métadonnées t_B et N_B , pour autant que les espaces vides entre deux filtres (par exemple : les mois de février, mars et avril dans la Figure 54) soient stockés dans le cube raster par des valeurs « no data ».

⁹⁶ Lorsqu'un filtrage horizontal est appliqué sur des dimensions continues (X , Y , et parfois B), le sous-cube résultant comprend toujours des valeurs « no data » pour modéliser les espaces vides entre les filtres de cette dimension. Le paramètre N_B du sous-cube inclut les filtres associés à une valeur « no data ».

Notons que lorsqu'un filtre est défini à partir d'un membre non-détaillé de B , tous les membres détaillés en relation hiérarchique avec le membre sélectionné par l'utilisateur sont considérés comme les filtres à appliquer sur le cube⁹⁷ (voir section 5.4.1).

5.7.2.2. Filtrage horizontal spatial

Le résultat d'un filtrage horizontal sur un membre spatial (dimensions R, C) de coordonnées ($R=r, C=c$) est un sous-cube comprenant tous les pixels du cube de coordonnées ($R=r, C=c$) dans le même ordre relatif que celui du cube. Un exemple de filtrage horizontal spatial à un filtre est donné dans la Figure 53b. Dans un sous-cube résultant d'un filtrage spatial à f filtres⁹⁸,

$$N_{R,C} = f$$

Puisque la hiérarchie spatiale est implicite, l'espace raster du sous-cube résultant est remis en relation avec l'espace géographique par simple modification des métadonnées N_R, N_C, t_R et t_C , pour autant que les espaces vides entre deux filtres spatiaux soient matérialisés par des valeurs « no data » (voir Figure 55).

Il est rare qu'un utilisateur définisse des filtres spatiaux directement par leurs coordonnées x, y . La plupart du temps, le filtrage horizontal spatial est défini sur base d'un ou plusieurs membre(s) spatialisé(s) tel qu'illustré à la Figure 55. Ce genre d'opération est assimilable à une opération locale d'algèbre de carte où le raster dimensionnel intégré (qui modélise les membres spatialisés) est un masque binaire appliqué sur le cube raster. Notons que si le raster dimensionnel a une emprise spatiale plus petite que le cube raster, l'emprise spatiale du cube raster est réduite pour correspondre à celle du raster dimensionnel au moment de son intégration. Cette réduction physique de la taille du cube est effectuée lors de l'opération de découpage expliquée à la section 5.5.3.4. Dans l'exemple de la Figure 55, le raster dimensionnel est déjà intégré et par conséquent, le raster dimensionnel et le cube raster partagent les mêmes métadonnées N_C et N_R (mais aussi t_R, t_C, h_R, h_C et SCR).

⁹⁷ Pour autant que la hiérarchie de B soit non-matérialisée. Si la hiérarchie de B est matérialisée, un filtrage horizontal sur B est toujours effectué sur le niveau le plus détaillé du cuboïde correspondant. Notons également qu'il est impossible d'effectuer un filtrage horizontal sur un niveau hiérarchique plus détaillé que celui du cuboïde sélectionné lors du filtrage vertical.

⁹⁸ Le paramètre $N_{R,C}$ du sous-cube inclut les filtres associés à une valeur « no data ».

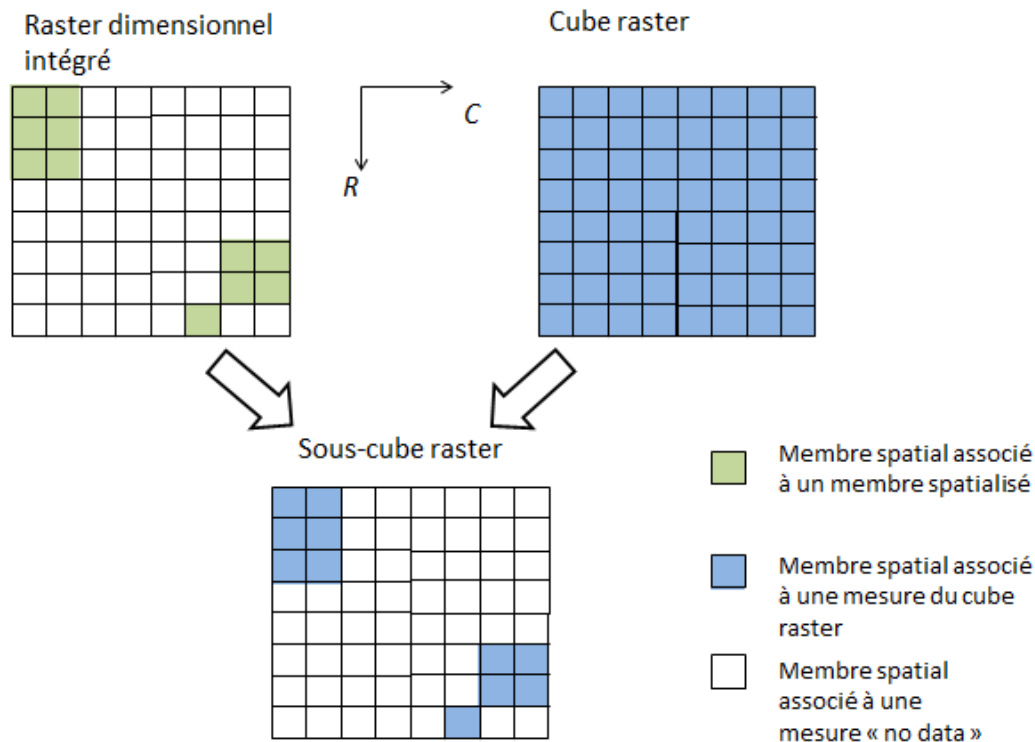


Figure 55. Application d'un filtre spatial à partir de membres spatialisés

5.7.3. Agrégation partielle

L'agrégation partielle est appliquée sur les dimensions dont la hiérarchie n'est pas matérialisée. Puisque dans notre modèle, nous considérons que la hiérarchie spatiale est toujours matérialisée, l'agrégation partielle ne peut avoir lieu que sur la dimension B^{99} si celle-ci est non-matérialisée (agrégation partielle non-spatiale), ou sur une couche de membres spatialisés importée à la volée (agrégation partielle spatiale). Comme nous allons le voir dans les prochaines sections, cette opération élémentaire est responsable de l'opération SOLAP de forage sur la dimension B (cas non-spatial) ou sur une couche de membres spatialisés (cas spatial).

5.7.3.1. Agrégation partielle non-spatiale

L'agrégation partielle non-spatiale dépend du niveau hiérarchique de B représenté dans l'interface graphique (voir section 5.8) et elle n'a de raison d'être que si ce niveau est non-détaillé. Si le niveau hiérarchique de B représenté dans l'interface est L ($L > 0$), le résultat de l'agrégation partielle est un cuboïde calculé à la volée où le nombre de membres de B est réduit au nombre de membres du niveau L (les opérations élémentaires suivantes sont donc appliquées sur ce cuboïde). D'après notre exemple de la section 5.6, la dimension B compte un niveau « mois » (niveau 0 ou niveau détaillé) et un niveau « trimestre » (niveau 1). Si l'interface représente les trimestres, l'agrégation partielle est appliquée sur un cube comptant 12 membres B (mois) et son résultat est un cuboïde

⁹⁹ Nous verrons que l'agrégation partielle est également appliquée sur d'autres dimensions (ROLAP) dans notre modèle relationnel (section 6.6.3.1).

comptant 4 membres B (trimestres). Tous les faits du cube sont donc redéfinis sur le niveau L de la dimension B et les mesures de ces nouveaux faits sont calculées à la volée. La Figure 56 est une représentation de cet exemple.

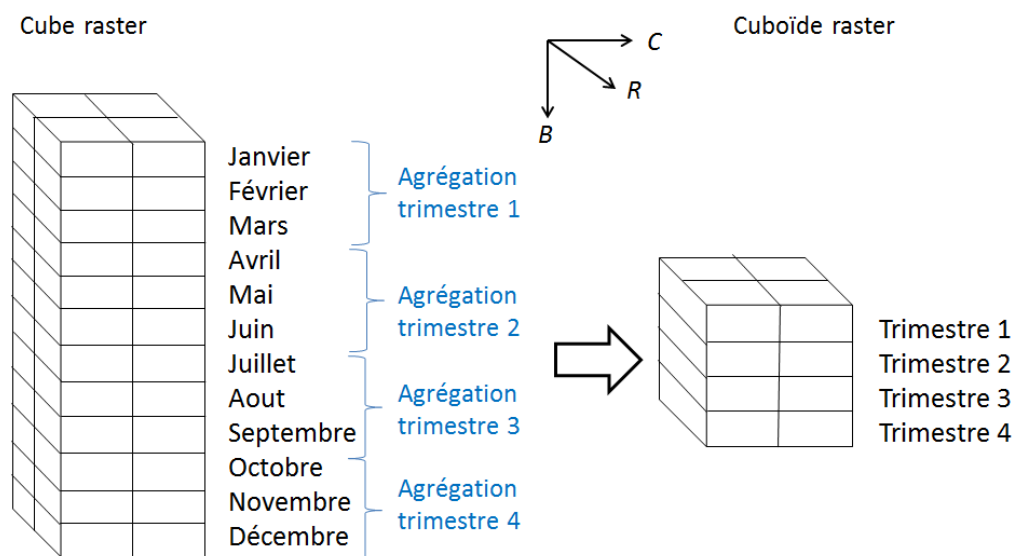


Figure 56. Agrégation partielle non-spatiale d'un cube raster

Le calcul d'une mesure d'un fait du cuboïde est une agrégation (par exemple : la somme) de toutes les mesures des faits détaillés en relation avec ce fait dans la définition hiérarchique de la dimension B (explicite pour une dimension quelconque, implicite pour une dimension continue) :

$$p_{b(L),r,c} = \sum p_{b(0),r,c}$$

où $b(0)$ est en relation avec $b(L)$ dans la hiérarchie de B

Si la hiérarchie de B est explicite (dimension quelconque), des pointeurs sont créés sur le niveau L pour établir la correspondance entre les valeurs d'index de B dans le cuboïde ($b(L)$) et leur correspondance sémantique (membre) dans la hiérarchie. Si la hiérarchie de B est implicite (dimension continue), la métadonnée h_B (homothétie de B) est simplement multipliée par le facteur d'échelle entre le niveau 0 et le niveau L (multipliée par 3 dans l'exemple de la Figure 56), et la métadonnée N_B est divisée par ce même facteur (3 dans l'exemple de la Figure 56).

La modification du paramètre de l'agrégation partielle (niveau de la dimension B affiché dans l'interface) correspond à une opération SOLAP de forage sur la dimension B lorsque sa hiérarchie est non-matérialisée (voir section 5.9.2.1).

5.7.3.2. Agrégation partielle spatiale (zonale)

L'agrégation partielle spatiale a lieu lorsque l'utilisateur souhaite afficher une couche de membres spatialisés dans une des interfaces. Puisque des membres spatialisés comptent des caractéristiques à la fois spatiales et sémantiques, cette opération est capable

d'alimenter aussi bien l'interface graphique que cartographique Elle correspond à une opération zonale d'algèbre de carte (voir section 2.3.1) qui calcule une mesure pour chaque fait défini à la fois par un membre spatialisé s d'une couche modélisée par un raster dimensionnel intégré, et par un membre b de la dimension B . Une mesure associée à un tel fait, notée $p_{s,b}$, est calculée par agrégation spatiale des mesures ($p_{r,c,b}$) de tous les faits du cube raster qui partagent les même coordonnées (r, c) que s . Si la fonction d'agrégation est la somme,

$$p_{s,b} = \sum p_{r,c,b} \text{ où } p'_{r,c} = s$$

Dans cette formule, la valeur d'un pixel de coordonnées ($R=r, C=c$) dans le raster dimensionnel intégré est notée $p'_{r,c}$ (voir section 5.5.1). Un exemple était donné dans la Figure 48.

Il y a deux méthodes possibles pour stocker le résultat de cette opération : une méthode vectorielle et une méthode raster. Dans la méthode vectorielle, qui permet une représentation dans l'interface graphique et cartographique, le résultat est un cuboïde à deux dimensions S (couche de membres spatialisés s) et B où chaque cellule est un élément du produit cartésien $S \times B$. Les coordonnées selon S sont des pointeurs renvoyant vers les identifiants des membres spatialisés de S . Pour être représentés dans l'interface cartographique, ceux-ci doivent être associés à des entités vectorielles décrivant leur géométrie. Un exemple de résultat d'agrégation partielle spatiale dans l'interface graphique selon l'approche vectorielle est donné dans la Figure 61.

Dans la deuxième méthode, qui n'exploite que le format raster, le résultat est un cuboïde raster de dimensions R, C, B où chaque pixel appartenant à un membre spatialisé s a comme valeur p_s . Cette technique implique cependant deux inconvénients majeurs. Le premier est dû aux éléments de la couche S qui ne sont plus identifiés sémantiquement dans le cuboïde résultant, et qui ne peuvent donc pas être affichés dans l'interface graphique¹⁰⁰, mais uniquement cartographique. Le second inconvénient provient de la redondance des mesures du cuboïde résultant puisque tous les pixels d'une même entité géographique s présentent la même valeur. Le cuboïde en sortie a d'ailleurs la même taille que le cube en entrée (ce qui n'est pas le cas de la première méthode qui réduit considérablement la taille du cube puisque les dimensions R et C disparaissent). Pour l'agrégation partielle spatiale, la méthode vectorielle semble donc préférable à la méthode raster. Un exemple de résultat d'agrégation partielle spatiale dans l'interface cartographique selon l'approche raster est donné dans la Figure 59.

Puisque les agrégations partielles sont responsables des opérations SOLAP de forage, on peut considérer que l'agrégation zonale décrite ici est responsable du forage vers un niveau d'une hiérarchie sémantique des dimensions spatiales X, Y (couche de membres spatialisés). Ce type de forage est très différent du forage spatial décrit à la section 5.3.2 (provoqué par le filtrage vertical puisque les niveaux spatiaux sont matérialisés) car ce

¹⁰⁰ A moins de réappliquer une opération spécifique pour retrouver ces valeurs sur base de raster dimensionnel en fin de chaîne.

dernier est basé sur une hiérarchie purement géométrique (et implicite) faisant varier la résolution spatiale du cube raster. Notons que si les niveaux de cette hiérarchie géométrique des dimensions X , Y n'étaient pas matérialisés dans notre modèle, un autre type d'agrégation partielle spatiale devrait être défini à partir des relations hiérarchiques implicites entre membres de différents niveaux spatiaux décrites à la section 5.3.2. Les opérations SOLAP de forage purement spatial (changement de résolution) seraient alors effectuées via cette opération élémentaire.

5.7.4. Agrégation totale

Les opérations élémentaires d'agrégation totale suppriment toutes les dimensions qui ne sont pas reprises dans l'interface. Dans notre modèle, cette interface est capable d'afficher soit un niveau¹⁰¹ des deux dimensions spatiales (carte), soit une couche de membres spatialisés (carte), soit un niveau d'une dimension non-spatiale (graphique), soit un niveau d'une dimension non-spatiale et une couche de membres spatialisés (graphiques). Pour plus de détails sur ces interfaces, nous renvoyons à la section 5.8.

Pour chaque fait affiché dans l'interface utilisateur, l'opération élémentaire présentée ici calcule une nouvelle mesure par agrégation totale des mesures du cube raster selon les dimensions non-reprises dans le fait. Une agrégation totale peut donc être définie comme une agrégation partielle appliquée sur le niveau « all » des hiérarchies des dimensions non-représentées dans l'interface (voir section 1.2.1). Le résultat d'une agrégation totale est un cuboïde où le nombre de membres (N) des dimensions non affichées par l'interface est réduit à 1.

Nous faisons une distinction entre les agrégations totales non-spatiales (selon la dimension B) appliquées sur les faits spatiaux (définis selon les dimensions R , C) et les agrégations totales spatiales (selon les dimensions R et C) appliquées sur les faits non-spatiaux (définis selon B). Cette différence est illustrée dans la Figure 57 et est expliquée dans les prochaines sections.

¹⁰¹ Par « afficher un niveau », nous entendons « afficher tous les faits définis par ce niveau ». Par exemple, afficher le niveau « mois » de la dimension temporelle équivaut à afficher le nombre de cambriolages (mesure) pour chaque mois de l'année et pour l'ensemble du territoire (faits).

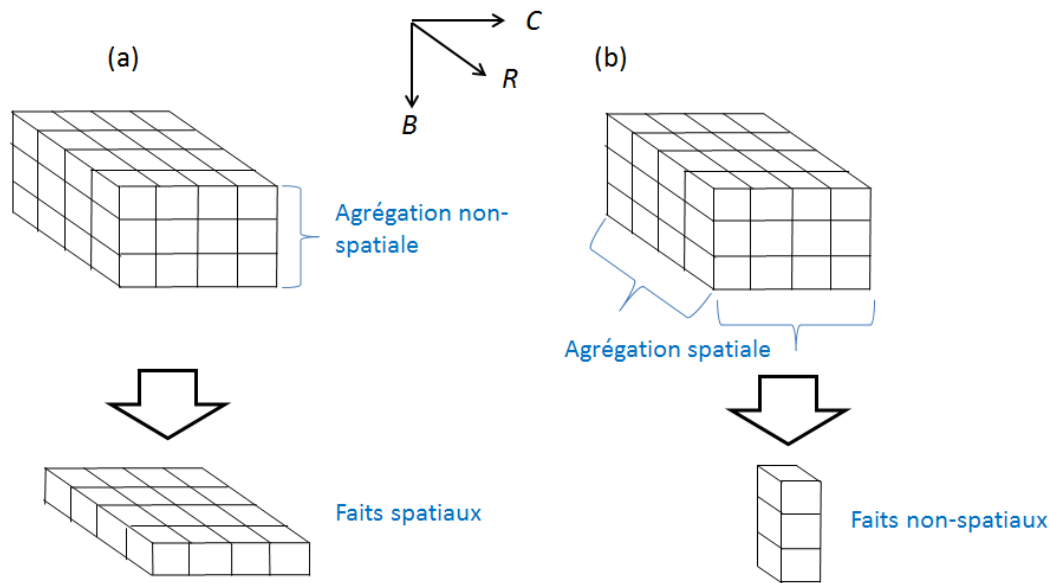


Figure 57. Agrégations totales non-spatiale et spatiale d'un cube raster

Remarquons que dans la Figure 52, nous considérons que l'agrégation partielle doit être appliquée avant l'agrégation totale car le résultat de l'agrégation partielle peut être exploité par l'agrégation totale pour alimenter à la fois les interfaces cartographiques et graphiques (voir section 5.8). Cela permet donc d'optimiser les opérations SOLAP appliquées à la fois sur la carte et sur le graphique. Il est cependant possible d'appliquer l'agrégation partielle après l'agrégation spatiale. L'intérêt est alors de simplifier l'opération d'agrégation partielle et donc d'optimiser les opérations de forage consécutives sur la dimension B dans l'interface graphique¹⁰² (voir section 5.9.2.1), ou l'affichage de membres spatialisés (voir sections 5.8.3 et 5.8.5).

5.7.4.1. Agrégation totale non-spatiale

L'agrégation totale non-spatiale d'un cube raster est appelée par l'affichage de faits spatiaux. Le résultat d'une telle agrégation est un raster où pour chaque fait associé à un membre de coordonnées ($R=r$, $C=c$), la mesure correspondante est l'agrégation de toutes les mesures du cube de coordonnées ($R=r$, $C=c$). Si l'agrégation est la somme :

$$p_{r,c} = \sum_{b=0}^{N_b-1} p_{b,r,c}$$

Un exemple d'agrégation totale non-spatiale d'un fait spatial est donné dans la Figure 57a. Notons que ce type d'opération correspond à une opération locale d'algèbre de carte (Tomlin, 1983, 1991).

Afin de ne pas alourdir notre formalisme, lorsque nous décrivons des opérations élémentaires d'agrégation (sections 5.7.4.1, 5.7.4.2 et 6.6.3), nous considérons toujours un cube raster de base dont la spatialité est décrite par R et C . Dans le cas particulier où le

¹⁰² En effet, l'agrégation partielle est alors appliquée sur un cube composé uniquement de la dimension B (tableau unidimensionnel).

cube raster résulte d'une agrégation zonale (partielle spatiale) selon l'approche vectorielle (voir section 5.7.3.2), les dimensions R , C doivent être simplement remplacées par S (couche de membres spatialisés). Dans le cas présent, la formule d'agrégation totale non-spatiale (appelée par l'affichage de faits spatialisés) devient :

$$p_s = \sum_{b=0}^{N_b-1} p_{b,s}$$

5.7.4.2. Agrégation totale spatiale

L'agrégation spatiale d'un cube raster est appelée par l'affichage de faits non-spatiaux. Pour chaque fait associé à un membre de coordonnée ($B=b$), la mesure correspondante est l'agrégation de toutes les mesures du cube de données de coordonnée ($B=b$). Si l'agrégation est la somme :

$$p_b = \sum_{r=0}^{N_R-1} \sum_{c=0}^{N_C-1} p_{b,r,c}$$

Un exemple d'agrégation spatiale d'un fait non-spatial est donné dans la Figure 57b.

5.8. Interfaces SOLAP raster

Comme nous l'avons vu, l'interface joue un rôle important dans les opérations SOLAP car elle détermine les faits pour lesquels une mesure doit être calculée par agrégation totale et partielle (voir section 5.7.3). Un SOLAP est généralement composé de trois types d'interfaces permettant de manipuler et d'afficher les faits d'un cube de données selon certaines dimensions : la table de pivot, le graphique et la carte. La table de pivot et le graphique sont directement hérités de l'OLAP alors que la carte est une interface supplémentaire spécifique au SOLAP.

5.8.1. La carte remplace la table de pivot

Dans notre modèle SOLAP raster à trois dimensions, la carte n'est pas considérée comme une interface supplémentaire aux deux autres, mais bien comme une interface remplaçant la table de pivot. Il ne s'agit pas d'un choix de modélisation délibéré, mais d'une conséquence de l'intégration de la spatialité directement dans le cube de données. En effet, la table de pivot est la représentation la plus fidèle d'un cube de données car elle affiche un tableau¹⁰³ de mesures organisées en rangées (membres d'une première dimension) et en colonnes (membres d'une seconde dimension). Les membres de dimension sont directement reconnus par l'utilisateur grâce à leur nom attribué à telle rangée ou telle colonne (par exemple : les produits en rangées et les magasins en colonnes). Dans le cas d'un cube raster, la représentation tabulaire n'a pas de sens puisque les membres des dimensions R et C n'ont pas de valeur sémantique, mais uniquement géométrique. L'utilisateur est donc capable de comprendre la signification des membres

¹⁰³ Dans cet exemple, on admet que la table de pivot représente deux dimensions. En pratique, une table de pivot peut représenter plus de deux dimensions.

spatiaux du cube raster uniquement via une carte représentant à la fois la dimension X et la dimension Y .

La représentation la plus fidèle d'un cube raster est donc bien une carte, et plus précisément une carte composée d'une couche raster. Une couche raster est en réalité un tableau superposable à n'importe quelle autre couche spatiale dans un même SCR. Au lieu d'afficher les valeurs numériques des cellules de tableau comme on le ferait dans une table de pivot¹⁰⁴, celles-ci sont uniquement associées à un code couleur renvoyant à une légende tout en respectant les règles de sémiologie cartographique. L'interface cartographique d'un SOLAP raster peut donc être considérée à la fois comme une carte et un tableau. Notons que le tableau raster est moins malléable que la table de pivot. Il est par exemple difficilement envisageable de représenter plusieurs niveaux hiérarchiques sur différents membres de dimensions avec un raster (cela impliquerait un raster présentant des niveaux de résolution spatiale différents d'un pixel à l'autre). De plus, l'opération OLAP « pivot », consistant à échanger la position des dimensions dans le tableau, n'a pas de sens dans un raster puisque la direction du Nord cartographique doit toujours pointer vers le haut (et non vers la droite).

5.8.2. Interface cartographique et dimensions spatiales

Le rôle principal de l'interface cartographique est de représenter un niveau hiérarchique des dimensions spatiales X et Y du cube raster. Ce niveau est un ensemble de faits déterminé par le produit cartésien $R \times C$ du cube résultant des différentes opérations de filtrage et d'agrégation partielle de la chaîne d'opérations élémentaires (voir section précédente). La carte est obligatoirement le résultat d'une agrégation totale non-spatiale (voir section 5.7.4.1) et représente un ensemble de mesures donné par :

$$\{p_{r,c} \mid 0 \leq r \leq N_R - 1, 0 \leq c \leq N_C - 1\}$$

La Figure 58 est un exemple de représentation des dimensions spatiales dans une interface cartographique SOLAP raster d'après le schéma en étoile de la section 5.6 : les cambriolages pour toute la période (année 2012) et tout le territoire couvert par le cube raster (Londres). Les cellules du raster ont comme valeur le nombre de cambriolages ayant eu lieu dans la surface délimitée par leur emprise spatiale. Elles sont associées à une intensité de rouge variant linéairement par rapport à la distribution de la mesure (le nombre de cambriolages). Les cellules de valeur nulle sont complètement transparentes. Le niveau de hiérarchie spatiale est « pixel = 1000 m ».

Dans toutes les illustrations de l'interface cartographique représentées dans les prochaines sections (à l'exception de la Figure 59), le raster issu du cube raster est superposé à un fond de carte WMS (en noir et blanc) provenant de données « Open Street Map » (Open Street Map, 2015).

¹⁰⁴ Certaines tables de pivot OLAP associent également un code couleur en plus de la valeur des cellules du tableau pour une meilleure vision synoptique.

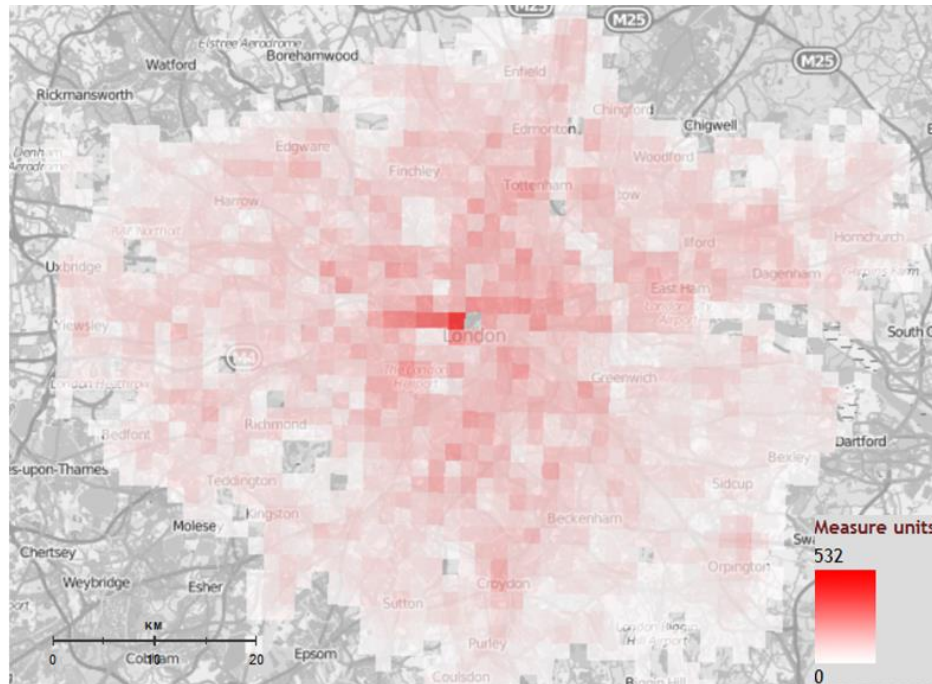


Figure 58. Représentation des dimensions spatiales dans l'interface cartographique d'un SOLAP raster

Pour obtenir le résultat de la Figure 58, la chaîne d'opérations élémentaires a été appliquée de cette manière :

- Filtrage vertical : cuboïde de niveau « pixel = 1000 m »
- Filtrage horizontal : aucun
- Agrégation partielle : aucune
- **Agrégation totale** : non-spatiale

5.8.3. Interface cartographique et membres spatialisés

Un autre rôle de l'interface cartographique est la représentation des membres spatialisés. Puisque ceux-ci sont constitués d'un ensemble de membres spatiaux (des dimensions X et Y), il est logique qu'ils puissent être gérés par cette interface. Dans ce cas, les faits représentés sont ceux définis par la couche de membres spatialisés¹⁰⁵ et leur mesure résulte obligatoirement d'une agrégation zonale (ou partielle spatiale) telle qu'expliquée à la section 5.7.3.2, et d'une agrégation totale non-spatiale (section 5.7.4.1). Si N_s est le nombre de membres spatialisés, l'ensemble de mesures représenté dans l'interface est donné par :

$$\{p_s \mid 0 \leq s \leq N_s - 1\}$$

La Figure 59 est un exemple de représentation d'une couche de membres spatialisés d'après notre schéma récurrent de cube raster : le nombre de cambriolages du mois de janvier 2012 par zone de police. Chaque pixel appartenant à une zone de police (membre

¹⁰⁵ Par exemple : les zones de forces de police.

spatialisé s) présente comme valeur le nombre de cambriolages se trouvant dans la zone¹⁰⁶. L'opération est effectuée sur le niveau spatial « pixel = 300 m ». Notons que certains pixels isolés présentent la valeur « no data » (couleur transparente) en raison de problèmes topologiques issus du fichier vectoriel rasterisé pour obtenir le raster dimensionnel stockant les membres spatialisés. Cette représentation résulte de la méthode raster décrite à la section 5.7.3.2. Le résultat de la méthode vectorielle est assez similaire, bien que moins « pixélisé ».

Remarquons aussi que cette représentation n'a pas été directement implémentée dans notre prototype SOLAP (voir section 8.3). Puisque son résultat se base sur une opération impliquant un raster dimensionnel et un raster issu de notre SOLAP, il est assez aisé d'importer le résultat de l'agrégation totale non-spatiale du cube dans un logiciel tiers pour y effectuer l'opération zonale d'algèbre de carte adéquate. Pour représenter les membres spatialisés, nous avons donc connecté le SIG-logiciel QGIS (2015) à notre entrepôt de données raster (lui-même géré par PostGIS). La symbolisation de la carte de la Figure 59, quant à elle, a été confiée au SIG-logiciel ArcGIS (ESRI, 2012). Techniquement, on peut dire que ce n'est donc pas le raster dimensionnel qui a été intégré au cube raster, mais le cube raster qui a été intégré au raster dimensionnel.

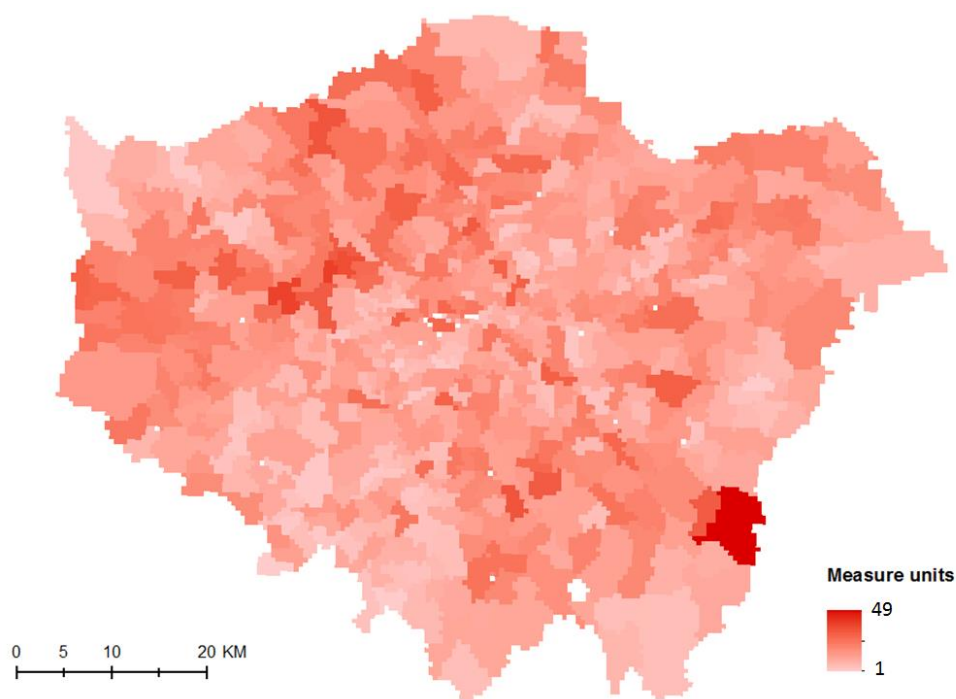


Figure 59. Représentation d'une couche de membres spatialisés dans l'interface cartographique d'un SOLAP raster

¹⁰⁶ Notons que dans cet exemple, nous nous sommes contentés de simplement agréger les mesures par la somme afin de rester cohérent dans l'ensemble de notre modélisation. Pour s'affranchir de l'influence de la surface des différentes zones de police dans la représentation de la mesure, il aurait été plus correct de représenter la densité de cambriolages par zone.

Pour obtenir le résultat de la Figure 59, la chaîne d'opérations élémentaires a été appliquée de cette manière¹⁰⁷ :

- Filtrage vertical : cube de base (niveau « pixel = 300 m »)
- Filtrage horizontal :
 - o Spatial : raster dimensionnel (zones de police)
 - o Non-spatial : membre « janvier 2012 »
- **Agrégation partielle :**
 - o **Spatiale** : couche « forces de police »
- **Agrégation totale** : non-spatiale

Notons que ce genre de représentation est typique des outils SOLAP vectoriels. Les agrégations y sont d'ailleurs moins complexes puisqu'elles sont traitées par membre spatial (que nous appelons « membre spatialisé ») et non par cellule. L'intérêt de proposer cette représentation dans un SOLAP raster provient donc essentiellement de sa capacité à intégrer les membres spatialisés à la volée.

5.8.4. Interface graphique et dimension non-spatiale

L'interface graphique est un bon complément à la carte car elle permet la représentation d'une dimension non-spatiale (B), alors que l'interface cartographique n'est capable de représenter que les dimensions X , Y et les membres spatialisés s . Un graphique affiche donc tous les faits définis par le niveau d'une dimension non-spatiale. Les mesures de ces faits résultent obligatoirement d'une agrégation totale spatiale et elles font partie de l'ensemble :

$$\{p_b \mid 0 \leq b \leq N_B - 1\}$$

La Figure 60 est un exemple d'interface graphique représentant l'évolution du nombre de cambriolages par mois¹⁰⁸ pour tout le territoire couvert par le cube raster (Londres).

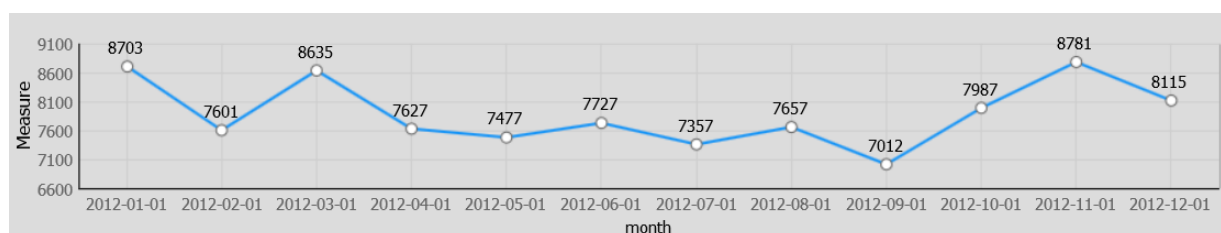


Figure 60. Représentation d'une dimension non-spatiale dans l'interface graphique d'un SOLAP raster

¹⁰⁷ En réalité, l'agrégation partielle spatiale a été appliquée après l'agrégation totale puisque cette opération a été réalisée par un logiciel tiers à la fin de la chaîne. Il nous semblait plus clair de décrire l'opération de manière cohérente par rapport à la chaîne de la Figure 52.

¹⁰⁸ Même si l'abscisse du graphique semble indiquer le premier jour du mois, les données sont bien agrégées sur tous les jours du mois. Ce problème est lié au prototype SOLAP raster servant à valider notre modèle.

Pour obtenir le résultat de la Figure 60, la chaîne d'opérations élémentaires a été appliquée de cette manière :

- Filtrage vertical : cuboïde de niveau « pixel = 1000 m »¹⁰⁹
- Filtrage horizontal : aucun
- Agrégation partielle : aucune
- **Agrégation totale** : spatiale

5.8.5. Interface graphique et membres spatialisés

Il est possible d'afficher une deuxième dimension dans l'interface graphique en exploitant plusieurs graphiques. La première dimension est représentée par la variation intra graphique et la deuxième dimension est représentée par la variation inter graphiques. Puisqu'un graphique ne peut représenter que des dimensions à connotation sémantique, seuls la dimension non-spatiale B et les membres spatialisés s d'une couche S sont exploitables par ce type d'interface dans notre modèle raster multidimensionnel. L'ensemble des faits représentés est donc celui déterminé par le produit cartésien $B \times S$, et les mesures résultent obligatoirement d'une agrégation partielle spatiale. L'ensemble de ces mesures est donné par :

$$\{p_{b,s} \mid 0 \leq b \leq N_B - 1, 0 \leq s \leq N_s - 1\}$$

La Figure 61 est un exemple de ce type de représentation. La variation intra graphique est le niveau « mois » de la dimension « temps » et la variation inter graphiques est une couche spatialisée dont les trois membres ont été digitalisés¹¹⁰ directement via l'interface cartographique. On peut facilement y comparer l'évolution du nombre de cambriolages (mesure) dans les trois quartiers de St-Giles, Shoreditch et Walworth.

Pour obtenir le résultat de la Figure 61, la chaîne d'opérations élémentaires a été appliquée de cette manière :

- Filtrage vertical : cuboïde de niveau « pixel = 300 m »
- Filtrage horizontal : aucun
- **Agrégation partielle** : spatiale
- **Agrégation totale** : aucune

Remarquons que ce genre de représentation aurait pu également convenir pour alimenter une interface tabulaire.

¹⁰⁹ Le filtrage vertical a sélectionné le cuboïde le moins détaillé sur la hiérarchie spatiale (pixel = 1000 m) afin d'optimiser l'opération d'agrégation totale spatiale calculant les mesures des faits affichés par le graphique.

¹¹⁰ Dans cette thèse, le terme « digitalisation » fait référence à la construction d'entités vectorielles en saisissant leurs coordonnées à l'écran.

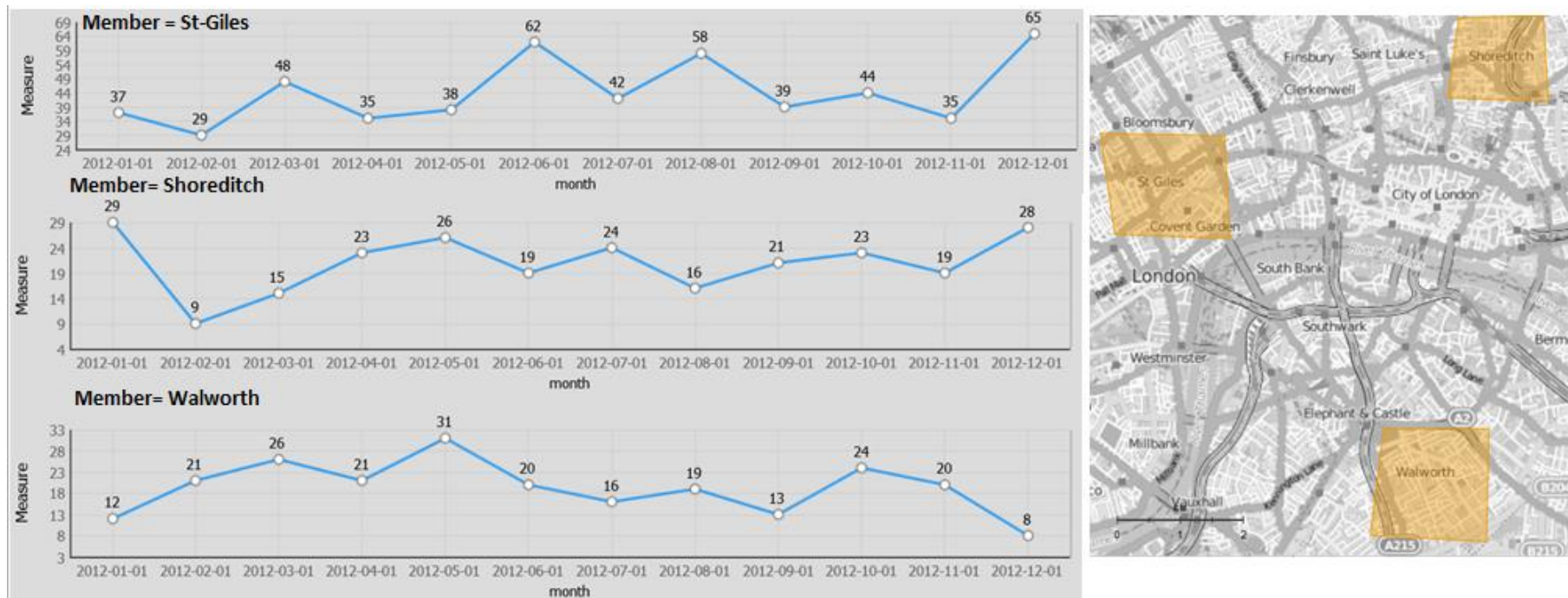


Figure 61. Interface graphique affichant une dimension B (mois : 12 membres) et trois membres spatialisés (quartiers londoniens)

Notons que dans la Figure 61, aucune dimension n'est affichée dans l'interface cartographique qui est là uniquement pour représenter les entités vectorielles digitalisées (dans le but de construire le raster dimensionnel modélisant la couche « quartiers de Londres »).

5.9. Opérations SOLAP

Les prochaines sections sont consacrées à la description des opérations SOLAP permettant à l'utilisateur de naviguer dans un cube raster à travers les interfaces présentées à la section 5.8. Ces opérations seront décrites au moyen de la chaîne d'opérations élémentaires présentée à la section 5.7 et schématisée dans la Figure 52. Chacune d'entre elles sera accompagnée d'un exemple issu du schéma en étoile présenté à la section 5.6.

Comme dans tout outil OLAP, ces opérations sont divisées en deux grandes catégories : les coupes et les forages. Comme dans tout outil SOLAP, ces catégories présentent elles-mêmes des caractéristiques différentes lorsqu'elles sont appliquées sur les dimensions spatiales (opérations spatiales) ou non-spatiales (opérations non-spatiales). Notons qu'une opération peut très bien être appliquée à la fois sur les dimensions spatiales et la dimension non-spatiale (opération mixte).

Dans la section 5.9.3, nous introduirons une catégorie supplémentaire d'opération SOLAP : les comparaisons. Bien que leur principe soit très simple, les informations qu'elles délivrent à l'utilisateur sont particulièrement intéressantes dans l'interface cartographique d'un SOLAP raster.

5.9.1. Opérations de coupe

Les opérations de coupe sont essentielles dans les outils OLAP car elles permettent à l'utilisateur d'observer la variation d'une mesure selon plusieurs dimensions (affichées dans l'interface) tout en fixant la définition des faits sur un ou plusieurs membres des autres dimensions (non-affichées dans l'interface). De plus, ces opérations impliquent un filtrage horizontal réduisant la dimensionnalité du cube (sous-cube) qui entraîne une simplification des agrégations (partielles et totales). En effet, le nombre de faits à traiter dans le sous-cube est inférieur à celui du cube de base. Sur les cubes de données volumineux (beaucoup de dimensions et/ou beaucoup de membres) comportant peu de hiérarchies matérialisées, l'utilisateur est donc obligé de jouer sur ces coupes pour que la durée des traitements reste acceptable (de l'ordre de la seconde).

Dans la chaîne d'opérations élémentaires de notre modèle raster, la coupe est effectuée au moment du filtrage horizontal. Si l'utilisateur décide d'appliquer une nouvelle coupe dans son analyse, la chaîne de traitement est donc reprise à l'étape de filtrage horizontal (sur le même cuboïde que pour l'opération SOLAP précédente). Lorsque le filtrage comprend un seul filtre par dimension, il s'agit d'une opération « slice ». Lorsque le filtrage comprend plusieurs filtres par dimension, il s'agit d'une opération « dice ».

5.9.1.1. Coupe non-spatiale

L'utilisateur peut appliquer une coupe non-spatiale en sélectionnant un ou plusieurs membres d'un niveau hiérarchique de la dimension B. La Figure 62 donne le résultat d'un exemple de coupe non-spatiale sur les interfaces cartographique et graphique. Les dimensions affichées sont les suivantes :

- Carte : dimensions spatiales X et Y (niveau « pixel=1000 m »)
- Graphique : dimension temporelle (niveau « mois »)

La coupe effectuée est un « slice » sur le premier trimestre de la dimension temporelle. Celui-ci inclut donc les mois de janvier, février et mars de l'année 2012. Le nombre de cambriolages représentés sur chaque cellule de la carte (mesure) est agrégé sur ce trimestre. Puisque le niveau hiérarchique des dimensions spatiales affiché est « pixel=1000 m », la coupe est réalisée sur le cuboïde correspondant à ce niveau. Le « slice » temporel aurait ainsi pu être appliqué directement après l'opération illustrée dans la Figure 58.

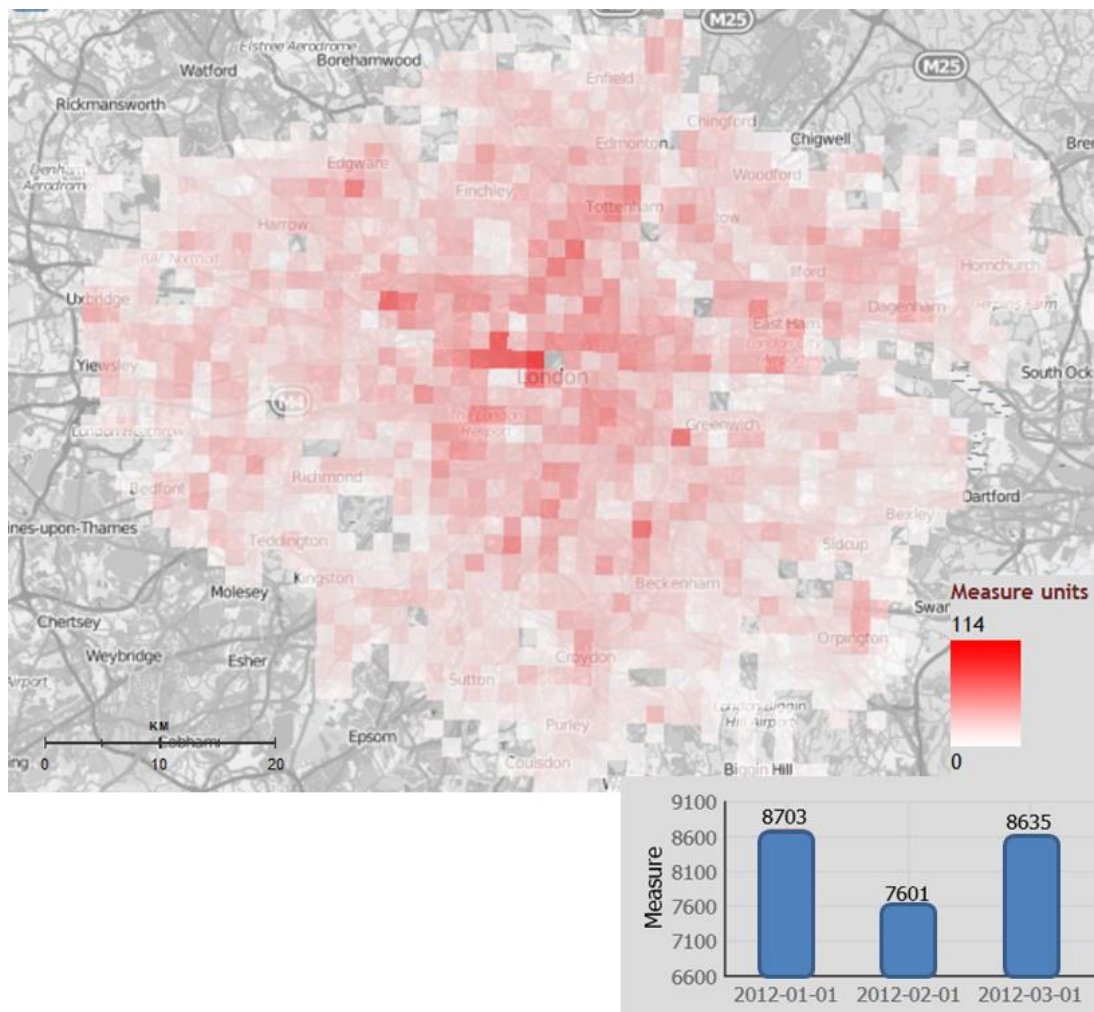


Figure 62. Résultat d'une coupe non-spatiale

Pour obtenir le résultat de la Figure 62, la chaîne d'opérations élémentaires a été appliquée de cette manière :

- Filtrage vertical : cuboïde de niveau « pixel = 1000 m »
- **Filtrage horizontal :**
 - o Non-spatial : membres « janvier 2012 », « février 2012 », « mars 2012 » (trimestre 1)
- Agrégation partielle :
 - o Non-spatiale : niveau « trimestre »
- Agrégation totale :
 - o Non-spatiale (carte)
 - o Spatiale (graphique)

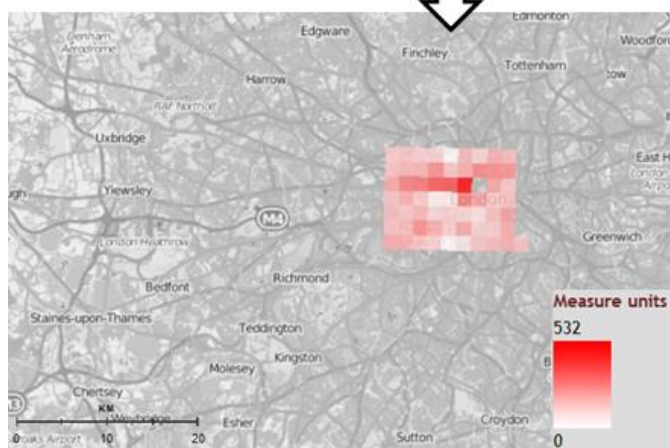
5.9.1.2. Coupe spatiale

Concrètement, l'utilisateur n'applique que très rarement une coupe directement sur les membres spatiaux (coordonnées x, y) du cube raster puisque nous avons vu à la section 5.5 que ces derniers n'ont aucune valeur sémantique. Pour que les membres spatiaux présentent une signification pour l'utilisateur, il est donc nécessaire de passer par une couche de membres spatialisés modélisée par un raster dimensionnel. Nous avons vu qu'un raster dimensionnel pouvait être intégré à partir d'une source quelconque (vecteur ou raster), mais il est également possible de créer un ou plusieurs membre(s) spatialisé(s) en digitalisant directement les zones d'intérêt sur l'interface cartographique (Figure 63). L'intégration du raster dimensionnel s'opère par rasterisation des entités vectorielles ainsi digitalisées, et ces entités modélisent les membres spatialisés créés à la volée par l'utilisateur. Une coupe spatiale est donc opérée par un filtrage spatial sur un ou plusieurs membre(s) spatialisé(s).

La Figure 63 illustre le résultat d'une coupe spatiale effectuée à partir d'un membre spatialisé digitalisé directement dans l'interface cartographique. Les dimensions X et Y (niveau « pixel = 1000 m ») sont affichées sur la carte et la dimension temporelle est affichée sur le graphique (niveau « mois »). La mesure est toujours le nombre de cambriolages sur les deux interfaces. Puisqu'aucune coupe non-spatiale n'est effectuée, les faits spatiaux de la carte représentent l'ensemble de la période temporelle (l'année 2012). La légende de la carte (étirement linéaire entre la valeur maximale et minimale de la carte) est adaptée à la zone couverte par la coupe spatiale. Les faits du graphique représentent chaque mois de l'année pour le centre de Londres.



Digitalisation d'un membre spatialisé
« Centre de Londres »



Rasterisation et intégration du
membre spatialisé

Coupe spatiale sur le centre de
Londres

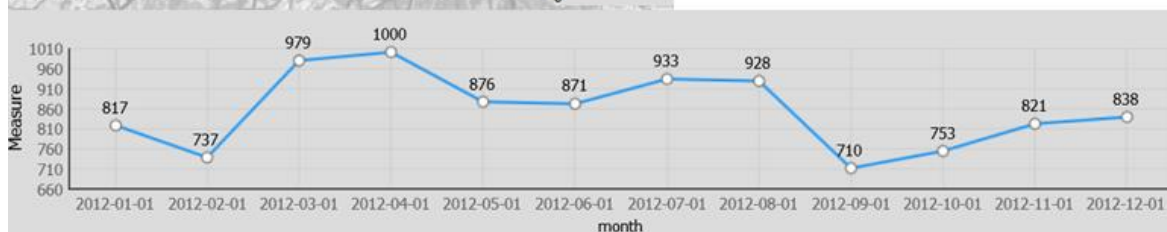


Figure 63. Création d'un membre spatialisé et résultat d'une coupe spatiale

Pour obtenir le résultat de la Figure 63, la chaîne d'opérations élémentaires a été appliquée de cette manière :

- Filtrage vertical : cuboïde de niveau « pixel = 1000 m »
- **Filtrage horizontal :**
 - o spatial : raster dimensionnel comportant un membre « centre de Londres »
- Agrégation partielle : aucune
- Agrégation totale :
 - o non-spatiale (carte)
 - o spatiale (graphique)

5.9.2. Opérations de forage

Les opérations de forage permettent à l'utilisateur de faire varier le niveau de granularité des mesures à travers les hiérarchies des dimensions affichées dans l'interface (cartographique ou graphique). Avec un « drill down », les mesures sont désagrégées à un niveau plus détaillé, et avec un « roll up », les mesures sont agrégées à un niveau moins détaillé. Ces opérations impliquent donc une redéfinition des faits affichés dans l'interface par rapport à la dimension forée (par exemple, affichage des trimestres à la place des mois dans l'interface graphique). Nous faisons une distinction entre les forages spatiaux (appliqués sur les dimensions spatiales) et les forages non-spatiaux (appliqué sur la dimension *B*). Ces deux types de forages seront détaillés dans les prochaines sections.

Le forage sur une dimension dont la hiérarchie est non-matérialisée est réalisé au moment de l'agrégation partielle dans la chaîne d'opérations élémentaires. Cela concerne donc éventuellement la dimension *B* puisque la hiérarchie spatiale est toujours matérialisée. Si l'utilisateur décide d'appliquer un nouveau forage sur une hiérarchie non-matérialisée dans son analyse, la chaîne de traitements est reprise à l'étape de l'agrégation partielle (sur le même sous-cube que pour l'opération SOLAP précédente).

Lorsqu'une hiérarchie est matérialisée par des cuboïdes (la hiérarchie spatiale et éventuellement, mais pas nécessairement, la hiérarchie de *B*), le forage sur cette hiérarchie est réalisé au moment du filtrage vertical dans la chaîne d'opérations élémentaires (première opération). Si l'utilisateur décide d'appliquer un nouveau forage sur une hiérarchie matérialisée dans son analyse, la chaîne de traitement est donc reprise à l'étape du filtrage vertical.

5.9.2.1. Forage non-spatial

Puisque le graphique est la seule interface capable d'afficher une dimension non-spatiale (*B*), un forage non-spatial ne peut être appliqué que sur cette interface¹¹¹. La Figure 64 est un exemple de forage sur une dimension temporelle (modélisée par *B*). L'opération « drill down » entraîne le passage du niveau « trimestre » au niveau « mois » et l'opération « roll up » entraîne le passage inverse. La mesure est le nombre de cambriolages et les faits concernent l'ensemble du territoire couvert par le cube (pas de coupe spatiale).

¹¹¹ A l'exception des forages sur les membres spatialisés qui peuvent être appliqués sur l'interface cartographique.

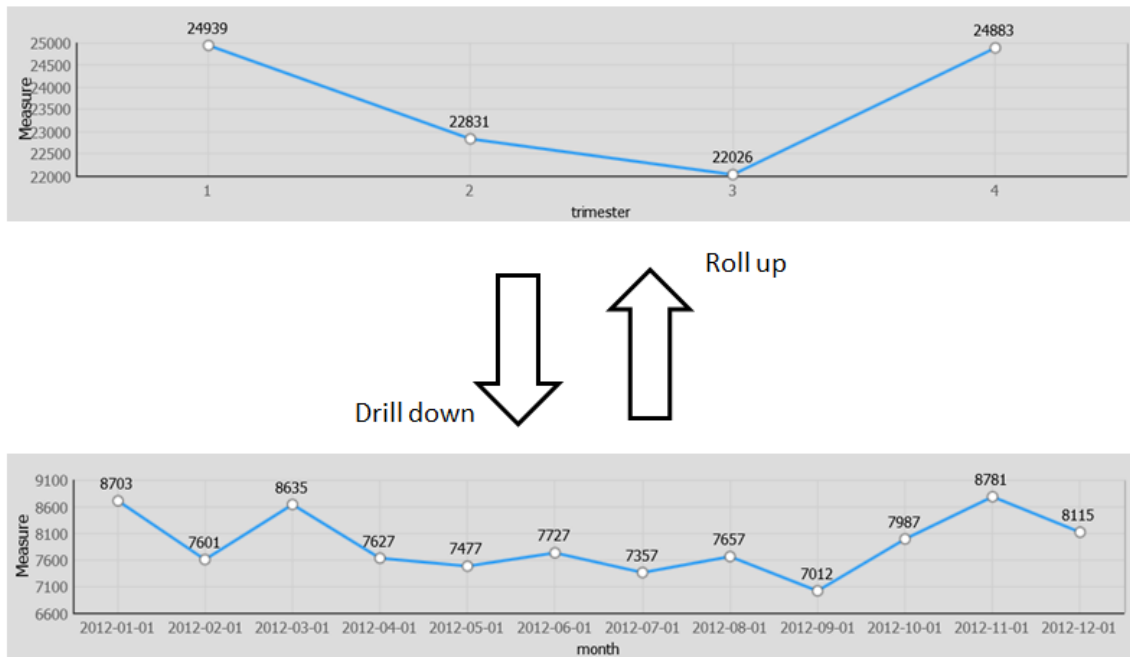


Figure 64. Forage non-spatial

Pour obtenir le résultat de la Figure 64, la chaîne d'opérations élémentaires a été appliquée de cette manière :

- Filtrage vertical : cuboïde de niveau « pixel = 1000 m »
- Filtrage horizontal : aucun
- **Agrégation partielle :**
 - o **Non-spatiale :** niveau « trimestre » pour l'opération « roll up », aucune pour l'opération « drill down »
- Agrégation totale : spatiale

5.9.2.2. Forage spatial

Puisque seule la carte est capable d'afficher les dimensions spatiales, un forage spatial est toujours appliqué sur cette interface. La Figure 65 en est un exemple : le « drill down » est un passage du niveau « pixel = 1000 m » au niveau « pixel = 300 m », et *vice versa* pour le « roll up ». La mesure est toujours le nombre de cambriolages et les mesures sont agrégées sur l'ensemble de la période (année 2012).

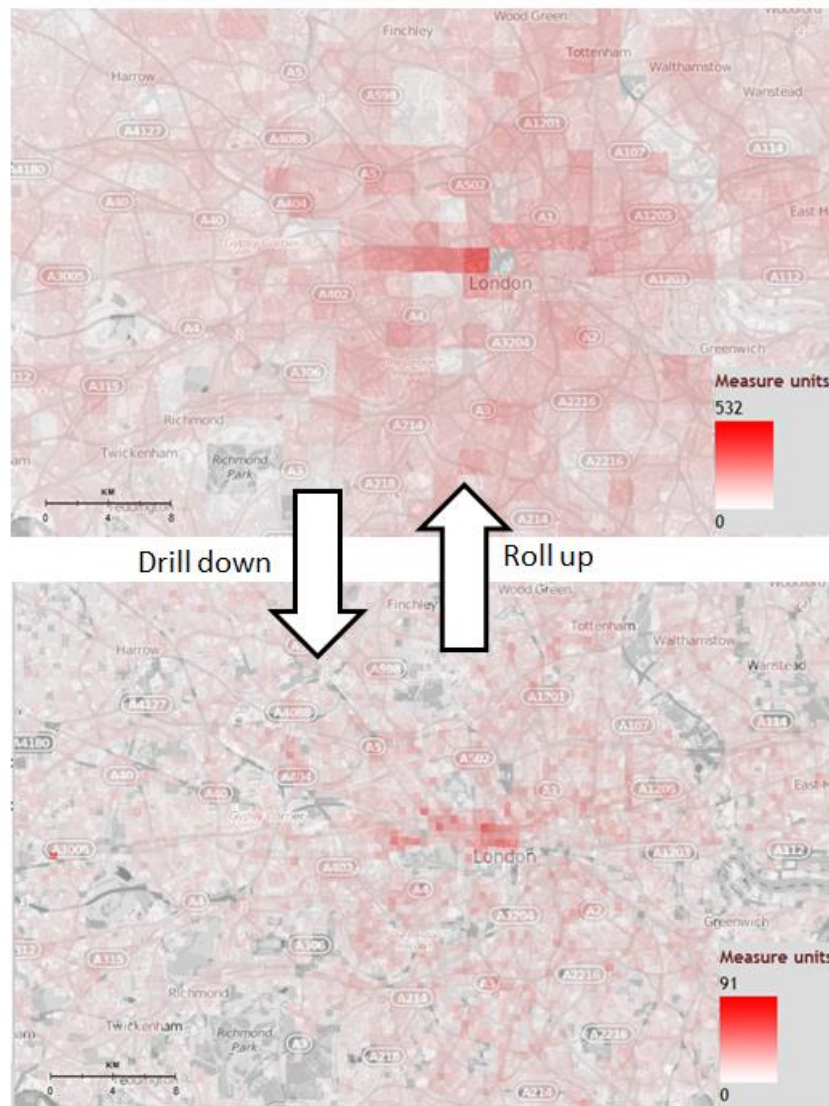


Figure 65. Forage spatial

Pour obtenir le résultat de la Figure 65, la chaîne d'opérations élémentaires a été appliquée de cette manière :

- **Filtrage vertical** : cuboïde de niveau « pixel = 1000 m » pour l'opération « roll up », cuboïde de niveau « pixel = 300 m » pour l'opération « drill down »
- Filtrage horizontal : aucun
- Agrégation partielle : aucune
- Agrégation totale : non-spatiale

5.9.3. Opérations de comparaison

Soient deux coupes d'un même cube¹¹², d'une même interface, et appliquées sur deux membres distincts (ou deux groupes distincts comptant le même nombre de membres) appartenant au même niveau hiérarchique d'une même dimension, une opération de comparaison est la différence entre les mesures des faits de la seconde coupe et les mesures des faits de la première coupe.

5.9.3.1. Comparaison cartographique

Lorsqu'une comparaison est effectuée sur l'interface cartographique, il s'agit d'une opération locale d'algèbre de carte consistant à calculer la différence entre deux raster. Puisque l'interface cartographique affiche les dimensions spatiales, une comparaison sur cette dernière n'a de sens que lorsqu'elle implique deux coupes non-spatiales.

La Figure 66 est un exemple de comparaison d'une mesure « nombre de cambriolages ». La Figure 66a est une coupe sur le premier trimestre 2012 et la Figure 66b est une coupe sur le second trimestre 2012. Ces deux coupes sont appliquées sur le même niveau hiérarchique des dimensions spatiales (même cuboïde) : « pixel = 1000 m ». La Figure 66c est le résultat d'une comparaison où la première coupe (a) est soustraite à la seconde coupe (b). Ce résultat est un raster montrant l'évolution du nombre de cambriolages entre le premier trimestre et le second semestre. Puisque les valeurs de pixel peuvent être négatives ou positives, la légende doit traduire une série divergente autour de la valeur 0. Dans l'exemple de la Figure 66, la partie positive (rouge) de la légende est un étirement linéaire entre la valeur maximale de pixel du raster et 0, et la partie négative de la légende (couleur bleue) est un étirement linéaire entre 0 et la valeur minimale de pixel du raster¹¹³. De cette manière, la carte de comparaison montre une évolution positive du nombre de cambriolages dans les zones rouges et une évolution négative dans les zones bleues.

L'intérêt de cette opération est de mettre en évidence de nouvelles zones géographiques en fonction de leur évolution positive ou négative, par rapport aux membres de coupe (filtres) de la dimension non-spatiale comparée.

¹¹² Par même cube, nous entendons le même cube de base, le même cuboïde (même niveau de forage) et le même sous-cube (mêmes coupes sur les autres dimensions).

¹¹³ La légende aurait également pu se baser sur la valeur absolue extrême la plus élevée pour son minimum et son maximum afin de mieux représenter les évolutions négatives et positives les unes par rapport aux autres.

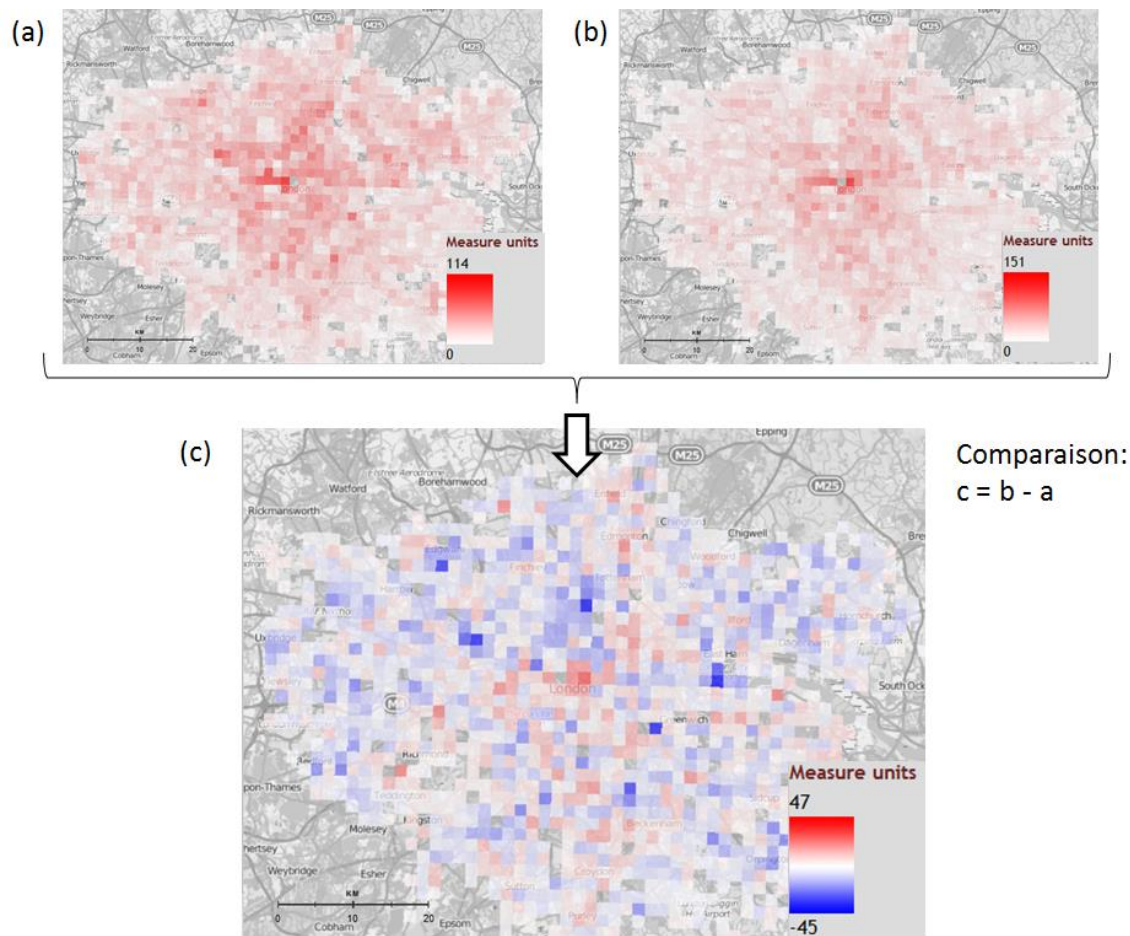


Figure 66. Comparaison entre deux coupes d'un cube raster

5.9.3.2. Comparaison graphique

Remarquons qu'il est également possible de comparer deux coupes spatiales sur l'interface graphique. Cette opération permet, par exemple, d'observer la différence du nombre de cambriolages entre deux quartiers de Londres sur un graphique affichant les membres « mois » de la dimension temporelle. La comparaison est simplement effectuée en appliquant une différence sur les valeurs des deux graphiques résultant des deux coupes spatiales respectives.

5.10. Entrelacement

Nous avons vu que physiquement, un tableau multidimensionnel est en réalité stocké comme une liste de valeurs dont la position est indexée selon une seule dimension. Dans un cube raster comme dans un cube OLAP tridimensionnel, il est donc nécessaire de faire la conversion entre l'espace conceptuel à 3 dimensions et l'espace physique à une dimension. Dans le cas le plus simple (tableau classique), l'index physique est donné par la formule suivante :

$$i_{b,r,c} = b * N_R * N_C + r * N_C + c$$

Cette formule nous montre bien que plus une dimension est positionnée loin dans l'ordre des indices du tableau, et plus la proximité de ses membres est conservée physiquement¹¹⁴. Dans ce dernier exemple, la dimension C , qui est exprimée en dernière position, est la dimension la plus favorisée puisque le passage d'un pixel d'une colonne (c) à la suivante ($c+1$) nécessite une incrémentation de 1. La dimension la moins favorisée est B (exprimée en première position) puisqu'elle nécessite une incrémentation de $N_R * N_C$ pour passer d'un membre (b) au membre suivant ($b+1$).

Dans le domaine du SIG raster, ce concept est appelé « entrelacement » (ESRI, 2012). L'entrelacement est la manière de physiquement organiser les données d'images satellitaires¹¹⁵ à plusieurs bandes. On parle d'entrelacement BSQ¹¹⁶ (Figure 67a) lorsque les dimensions raster sont exprimées dans l'ordre $i_{b,r,c}$ qui favorise la proximité des colonnes, puis des rangées, au détriment des bandes (cas le plus courant). L'entrelacement BIP¹¹⁷ (Figure 67b) exprime les dimensions raster dans l'ordre $i_{r,c,b}$ pour favoriser la proximité des bandes, puis des colonnes, au détriment des rangées. Enfin, il existe également l'entrelacement BIL¹¹⁸ (Figure 67c) qui exprime les dimensions raster dans l'ordre $i_{r,b,c}$ pour favoriser la proximité des colonnes, puis des bandes, au détriment des rangées. Le choix de l'entrelacement repose sur l'optimisation de certains traitements exploitant davantage les dimensions spatiales (rangée et colonne) ou la bande.

¹¹⁴ Dans un stockage idéal, deux pixels qui sont proches en trois dimensions devraient être proches dans la liste (et *vice versa*). Or, dans ce dernier exemple, deux cellules qui sont proches en trois dimensions peuvent se retrouver très éloignées dans la liste. Par exemple, les pixels (0, 0, 0) et (1, 0, 0) sont des voisins en trois dimensions, mais leurs index diffèrent de $N_R * N_C$. Inversement, les index des pixels (0, 0, $N_C - 1$) et (0, 1, 0) ne diffèrent que de 1, même si ces pixels ne sont pas des voisins en trois dimensions.

¹¹⁵ Ou de manière plus générale, les raster exploitant une dimension « bande ».

¹¹⁶ « Band SeQuential ».

¹¹⁷ « Band Interleaved by Pixel ».

¹¹⁸ « Band Interleaved by Line ».

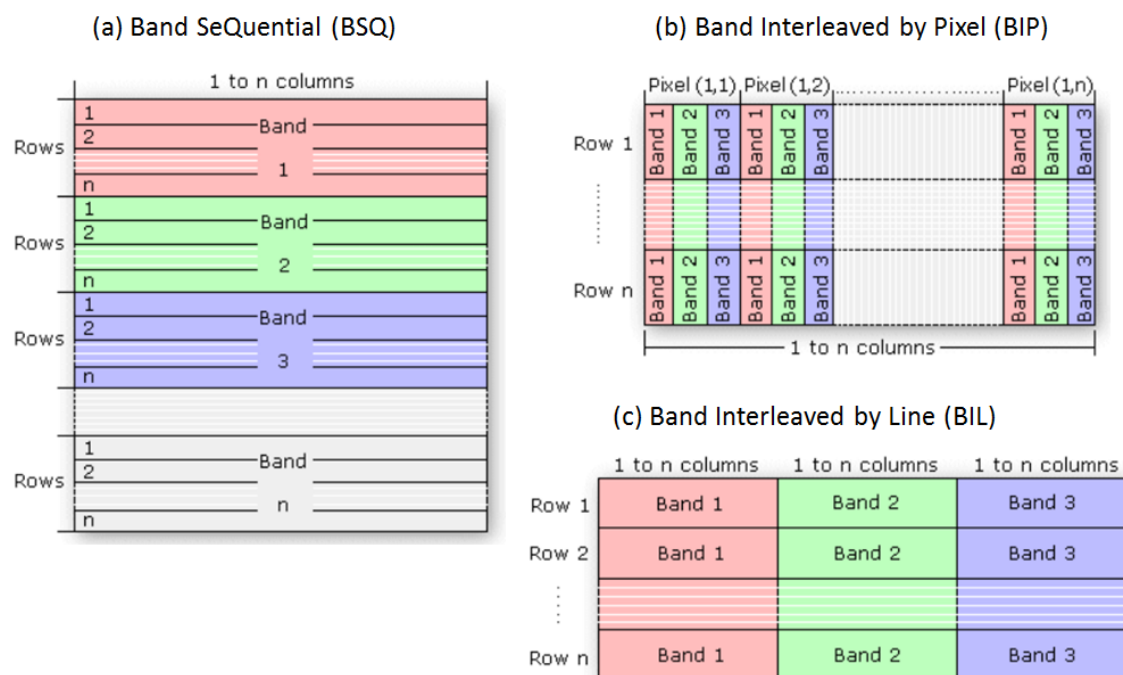


Figure 67. Les trois types d'entrelacements d'un cube raster (ESRI, 2012)

Quel entrelacement choisir pour implémenter notre modèle de cube raster? A priori, cette question n'aurait pas de réponse absolue puisque ces méthodes d'indexation favorisent toujours certaines opérations OLAP au détriment d'autres. BSQ (b, r, c) optimise les agrégations spatiales et BIL (r, c, b) optimise les agrégations non-spatiales. BIP (r, b, c) apparaîtrait donc comme un compromis entre les deux méthodes précédentes. Cependant, il peut sembler étrange d'éloigner physiquement des dimensions (R et C) que nous considérons comme étroitement liées d'un point de vue conceptuel (par la définition de membres communs). En effet, sauf dans un cas fortuit où un membre spatialisé d'un raster dimensionnel serait défini uniquement sur les cellules d'une même rangée, une coupe spatiale fait toujours intervenir les dimensions R et C de manière globalement équivalentes. Nous rejetons donc l'option BIP.

Dans la section 5.3.2, nous avons vu qu'une hiérarchie définie sur les dimensions spatiales pouvait être matérialisée efficacement par des cuboïdes (pyramide raster). Or, les opérations de forage (« roll up » ou « drill down ») entre niveaux hiérarchiques ainsi pré-calculés ne nécessitent pas d'agrégations des mesures sur les membres de cette dimension. Dans l'absolu, cette optimisation hiérarchique des dimensions spatiales donnerait donc un léger avantage à l'entrelacement BIL puisque une partie des opérations d'agrégations spatiales sont déjà pré-calculées. Cependant, les agrégations spatiales interviennent aussi dans les représentations graphiques et dans la représentation cartographique de membres spatialisés, et il est possible de matérialiser des cuboïdes sur la dimension B.

Il n'y a donc pas de préférence absolue entre l'entrelacement BSQ et BIL. Le choix entre ces deux méthodes dépend de la complexité relative des opérations d'agrégation sur la dimension B et sur les dimensions spatiales dans le cube de données. La complexité d'une opération d'agrégation est directement liée au nombre de membres à agréger. Ainsi,

si le nombre total de membres détaillés spatiaux du cube ($N_{R,C}$) est supérieur au nombre total de membres détaillés non-spatiaux (N_B), on peut admettre que les agrégations spatiales sont globalement plus complexes que les agrégations non-spatiales. Par conséquent, un entrelacement BSQ est préférable dans ce cas, et un entrelacement BIL est préférable dans le cas inverse. Lorsqu'une hiérarchie spatiale est matérialisée par des cuboïdes, il est également envisageable d'utiliser un entrelacement BIL pour les cuboïdes de niveaux spatiaux plus grossiers (et donc, comportant moins de membres spatiaux), et un entrelacement BSQ pour les niveaux spatiaux plus fins (comportant plus de membres spatiaux).

5.11. Conclusion

Nous venons de décrire toutes les caractéristiques d'un modèle SOLAP exploitant le format raster bien connu du domaine du SIG. Pour établir ce modèle, nous sommes partis de la définition d'un cube MOLAP comportant trois dimensions : R , C et B . En associant des métadonnées spatiales aux dimensions R et C (espace raster), nous avons démontré que celles-ci pouvaient modéliser des dimensions planimétriques X et Y implicitement, de la même manière que dans la modélisation raster propre au SIG. Nous avons donc simplement appelé ce type de cube « cube raster ». Les cellules de ce cube (pixels), sont considérées comme des faits spatiaux auxquels est associée une mesure, et un membre spatial est à la fois défini par la dimension R et la dimension C . En nous inspirant des pyramides raster, nous avons démontré qu'une hiérarchie implicite pouvait être définie sur ces dimensions pour modéliser différents niveaux de résolutions spatiales. Afin d'optimiser les opérations SOLAP de forage sur cette hiérarchie (forage spatial), les niveaux sont stockés dans des cuboïdes raster.

Après avoir discuté des caractéristiques des dimensions spatiales R et C , nous avons défini les caractéristiques de la dimension B , habituellement associée à la bande spectrale dans la modélisation raster. Dans notre modèle, celle-ci peut être associée à n'importe quelle dimension¹¹⁹ comme dans tout modèle OLAP. Nous avons cependant souligné le fait qu'il était intéressant de l'associer elle aussi à des métadonnées lui permettant de modéliser implicitement une dimension continue telle que le temps ou la hauteur. Notons que si notre travail s'était limité à un modèle tridimensionnel, la dimension B devrait nécessairement modéliser le temps puisqu'un entrepôt de données est un outil d'archivage (les deux autres dimensions étant obligatoirement des dimensions spatiales).

Puisque la définition des dimensions spatiales X , Y est implicite grâce aux métadonnées du cube raster, n'importe quel raster externe partageant les mêmes métadonnées que le cube peut être mis en relation avec ce dernier pour intégrer une couche de membres spatialisés S dans l'analyse de l'utilisateur. Ce que nous appelons « membre spatialisé » est l'équivalent d'un membre d'une dimension spatiale dans la littérature SOLAP au sens large (par exemple : des communes, des routes, *etc.*). Dans notre modèle, il aurait été ambigu d'utiliser le terme « dimension spatiale » puisqu'il fait déjà référence à X et Y . De plus, les membres spatialisés (ou membres géographiques) n'étant pas définis dans le schéma conceptuel en étoile (intégration à la volée), ils ne sont rattachés à aucune

¹¹⁹ A l'exception des dimensions spatiales planimétriques qui sont déjà modélisées par R et C .

dimension prédéfinie comme c'est le cas dans un SOLAP vectoriel. Un raster modélisant ainsi une couche de membres spatialisés est appelé « raster dimensionnel », et lorsque ce raster partage les mêmes métadonnées que le cube, nous l'appelons « raster dimensionnel intégré ». Les opérations nécessaires à cette intégration à la volée dans l'analyse de l'utilisateur ont été décrites de manière précise.

Après avoir illustré notre cube raster par un exemple concret impliquant des données de criminalité¹²⁰ de la ville de Londres, nous avons décrit une chaîne d'opérations élémentaires (agrégations et filtrages) permettant le calcul des mesures des faits affichés dans une interface cartographique ou graphique (la table de pivot n'a pas de raison d'être dans notre modèle puisqu'une cellule détaillée est un pixel raster nécessitant un positionnement sur une carte pour être interprété par l'utilisateur). En particulier, les fonctions d'agrégation sont directement inspirées de l'algèbre de carte et les filtrages exploitent des découpages de tableaux bien connus du domaine raster. Dans les interfaces, l'utilisateur a la possibilité de visualiser tous les types de dimensions (spatiales et non-spatiales) ainsi que les membres spatialisés, et d'exécuter les opérations SOLAP classiques sur un cube raster : coupes, forages, mais aussi comparaisons. Toutes ces opérations présentent une variante spatiale et non-spatiale. Elles ont été décrites au moyen de la chaîne d'opérations élémentaires et ont été illustrées dans le cube raster londonien présenté précédemment. La dernière section de ce chapitre était consacrée à l'indexation physique d'un cube raster qui présente exactement les mêmes caractéristiques que les données raster (entrelacement).

En conclusion, ce modèle raster exploite la spatialité des faits directement dans la structure multidimensionnelle d'un cube de données à travers le format raster et valide donc en partie notre hypothèse de recherche (considération du raster comme un cube MOLAP). De plus, sa capacité à intégrer n'importe quelle couche de membres spatialisés à la volée lui donne un avantage par rapport à la plupart des modèles SOLAP vectoriels. Le nombre de dimensions non-spatiales du modèle raster multidimensionnel est cependant limité à une seule (nous n'avons pas connaissance de raster à plus d'une dimension « bande » et les cubes MOLAP sont eux-mêmes très limités par le nombre de dimensions). Dans le prochain chapitre, nous définirons une extension de notre modèle raster SOLAP qui est capable de gérer un nombre indéfini de dimensions non-spatiales : le modèle raster relationnel.

¹²⁰ Notons que lorsque nous utilisons le terme « criminalité », nous faisons davantage référence au terme anglais « crime » qu'à son homonyme français. La « criminalité » englobe donc tous les types de délits.

Chapitre 6 : modèle raster relationnel

6.1. Introduction

Au chapitre précédent, nous avons présenté un modèle SOLAP raster où les dimensions spatiales planimétriques X , Y sont physiquement intégrées dans la structure multidimensionnelle d'un cube de données. Ce modèle a été établi sur base de la notion de cube raster qui exploite les nombreux points communs entre les cubes MOLAP et les données raster issues du domaine du SIG. Il hérite donc des qualités du MOLAP (performance des requêtes accrue par l'indexation des membres dimensionnels), mais aussi de ses principaux défauts (voir section 1.3.5.2) :

- Une mauvaise gestion de la faible densité : toutes les combinaisons de faits détaillés possibles occupent une place dans le cube, même si aucune mesure ne leur est associée (mesure « no data » ou 0).
- Un nombre limité de dimensions : seules trois dimensions peuvent¹²¹ être stockées dans le cube, dont deux sont obligatoirement les dimensions spatiales planimétriques X et Y . Par contre, le nombre de membres spatialisés est illimité et n'affecte pas les performances du SOLAP puisque ces membres peuvent être intégrés à la volée sous forme de nouvelles couches.
- L'absence d'une véritable norme¹²² pour les outils capables de supporter ce type de modèle: de nombreux SIG-logiciels utilisent le format raster, mais aucun ne les assimile à un entrepôt de données (et son formalisme multidimensionnel), et aucun outil SOLAP actuel n'est capable d'intégrer des données raster.

Dans ce chapitre, nous présentons l'extension relationnelle de notre modèle multidimensionnel raster. Celle-ci n'est pas une alternative au modèle du chapitre précédent, mais bien un complément permettant d'apporter une solution aux trois problèmes posés ci-dessus. L'association de ces deux modèles (raster et relationnel) présente à la fois des caractéristiques multidimensionnelles (comparables à du MOLAP) et relationnelles (comparables à du ROLAP). Contrairement à ce que l'on pourrait penser au premier abord, il ne s'agit pas d'une architecture « Hybrid OLAP » (voir section 1.3.5.3) car l'association entre multidimensionnel (raster) et relationnel y est très différente de celle observée dans la littérature. Dans une architecture hybride classique¹²³, l'utilisateur exploite des cubes de données MOLAP (magasins de données) construits à partir d'un entrepôt de données relationnel (ROLAP). Dans notre architecture, les cubes de données raster sont directement considérés comme mesures dans un schéma ROLAP (voir section suivante).

¹²¹ Théoriquement, un plus grand nombre de dimensions peut être stocké, mais cela affecte négativement les performances du cube.

¹²² Nous avons vu que les outils supportant le MOLAP étaient pratiquement tous propriétaires et qu'ils ne s'appuyaient pas tous sur une norme solide comparable au SQL des SGBD relationnels.

¹²³ Ce type d'architecture, basé sur des magasins de données, sera exploité dans le chapitre 8.

Le modèle SOLAP raster décrit dans ce chapitre se base sur une architecture ROLAP qui présente plusieurs intérêts. Le premier est la possibilité de l'implémenter dans un SGBD relationnel si celui-ci offre les fonctionnalités de base liées aux données raster. Nous avons présenté deux SGBD de ce type à la section 2.4 : Oracle GeoRaster (2012) et PostGIS (Racine & Cumming, 2011). Notre modèle s'appuie ainsi sur une technologie relationnelle robuste. Le deuxième intérêt est qu'il permet d'augmenter le nombre de dimensions non-spatiales du modèle SOLAP purement raster (qui était jusqu'à présent limité à une seule dimension non-spatiale). En comptant les deux dimensions spatiales et le nombre indéfini de dimensions non-spatiales, une instance du modèle « raster relationnel » est donc un « hypercube raster ». Finalement, nous verrons que ce modèle ROLAP apporte une réponse partielle à la gestion de la faible densité des données raster.

Ce chapitre est structuré de la façon suivante. Comme nous avons défini la notion de cube raster à trois dimensions (R , C , B) au chapitre précédent, nous allons commencer par définir un hypercube raster comportant un nombre indéfini de dimensions (section 6.2). Cet hypercube étant caractérisé par une composante raster (chapitre précédent) et une composante relationnelle (présent chapitre), nous allons clairement différencier les éléments de sa structure multidimensionnelle (fait, mesure, dimension, membre, hiérarchie et cuboïde) selon ces deux aspects (section 6.3). Les contraintes (métadonnées) appliquées sur les mesures relationnelles, ou mesures ROLAP, pour optimiser les opérations du modèle raster seront ensuite établies dans la section 6.4. La première partie de ce chapitre sera clôturée par la présentation d'un exemple d'hypercube raster (section 6.5) exploitant des données londoniennes avec une dimension non-spatiale supplémentaire (les types de délits) par rapport à l'exemple de cube raster du chapitre précédent. Cet exemple sera régulièrement utilisé pour illustrer les concepts présentés dans la deuxième partie.

Dans la deuxième partie de ce chapitre, nous décrirons une chaîne d'opérations élémentaires permettant le calcul des mesures affichées dans une interface du modèle raster relationnel (section 6.6). Cette chaîne est composée à la fois d'opérations propres au modèle purement raster (modèle multidimensionnel, voir chapitre précédent), et au modèle ROLAP. Puisque les interfaces et les opérations SOLAP permettant à l'utilisateur de manipuler un hypercube raster sont sensiblement les mêmes que celles du modèle raster, une courte section (6.7) sera consacrée à cet aspect. Nous discuterons ensuite de l'intérêt de la dimension raster B dans le modèle ROLAP. Ce chapitre sera naturellement finalisé par une conclusion dans la section 6.9.

6.2. Hypercube raster

Un hypercube raster est une instance du schéma conceptuel multidimensionnel représenté à la Figure 68. Ce schéma en étoile représente une mesure reliée à trois dimensions raster (voir section 5.2) et n dimensions relationnelles ($n \in \mathbb{N}$). Un fait détaillé est un élément du produit cartésien des $n+3$ dimensions auquel est associée une mesure atomique, et un fait est un ensemble de faits détaillés pour lequel une mesure est calculée par agrégation des mesures des faits détaillés qui le composent.

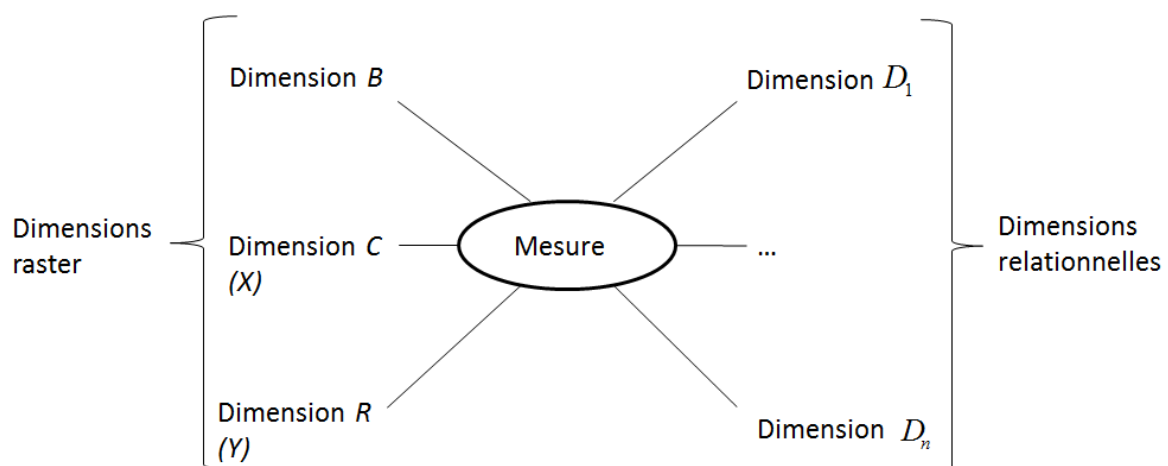


Figure 68. Schéma conceptuel d'un hypercube raster

Conceptuellement, une dimension relationnelle (ou ROLAP) n'est pas différente d'une dimension raster. La différence se situe au niveau logique. Alors qu'un membre d'une dimension raster est identifié par l'index d'un cube raster (ou deux index dans le cas des dimensions spatiales), un membre d'une dimension relationnelle est identifié par la clé primaire d'une table. Il peut s'agir d'une table des faits si la dimension est dégénérée, d'une table de dimension si le schéma est en étoile, ou d'une table de niveau dimensionnel si le schéma est en flocon (voir section 1.2.4). Une dimension relationnelle correspond donc à la dimension d'une architecture ROLAP telle que définie par Kimball & Ross (2013). Pour que les dimensions relationnelles puissent être mises en relation avec les dimensions raster, le schéma conceptuel de la Figure 68 est implémenté par un modèle logique tel que représenté à la Figure 69.

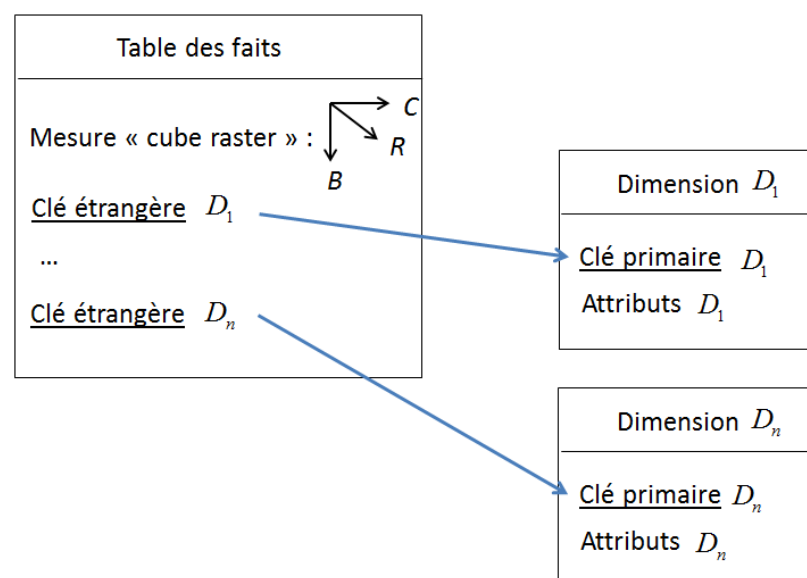


Figure 69. Schéma logique d'un hypercube raster

Le schéma de la Figure 69 est en étoile, mais tout autre type de schéma multidimensionnel ROLAP est également valable (flocon, constellation ou galaxie) selon les besoins de l'utilisateur. Dans tous les cas, la table des faits établit la relation entre les dimensions relationnelles et la mesure relationnelle qui n'est autre qu'un cube raster, lui-même composé des dimensions R , C et B . Pour chaque fait détaillé défini par le produit cartésien des dimensions relationnelles, il existe donc théoriquement un cube raster de dimensions R , C , B . En pratique, si une combinaison de dimensions relationnelles est associée à un raster vide, il n'y a tout simplement pas d'enregistrement associé à ce fait dans la table des faits. Ce modèle ne souffre donc pas du problème de la faible densité pour les dimensions relationnelles.

6.3. Structure multidimensionnelle

Comme nous venons de le voir, une fois implémenté par un hypercube raster, le schéma conceptuel de la Figure 68 est éclaté en deux modèles logiques : le modèle raster (cube raster) et le modèle ROLAP (cube¹²⁴ ROLAP). Il est donc important de bien différencier les notions suivantes :

- Cube raster et cube ROLAP
- Dimension raster et dimension ROLAP
- Membre raster et membre ROLAP
- Fait raster et fait ROLAP
- Mesure raster et mesure ROLAP
- Hiérarchie raster et hiérarchie ROLAP
- Cuboïde raster et cuboïde ROLAP

Les prochaines sections ont pour but de clarifier ces dualités et d'identifier l'impact du modèle raster dans le modèle ROLAP.

6.3.1. Fait et mesure

Le fait d'un des deux modèles (raster ou ROLAP) est toujours limité aux dimensions faisant partie de ce modèle. Ainsi, les faits raster sont définis uniquement par les dimensions raster (R , C , B) et les faits ROLAP sont définis uniquement par les dimensions ROLAP. Un fait (ou fait global) est la concaténation d'un fait raster et d'un fait ROLAP.

Une mesure raster atomique est une valeur d'un pixel et une mesure ROLAP atomique est un cube raster. La mesure associée à un fait raster non-détaillé est la valeur d'un nouveau pixel résultant de l'agrégation des pixels (ou mesures raster) d'un cube raster telle que définie à la section 5.7.3, et la mesure associée à un fait ROLAP non-détaillé est un nouveau cube raster résultant de l'agrégation d'un ensemble de cubes raster. Cette opération, comparable à une opération de « drill across » (voir section 1.3.2), sera détaillée aux sections 6.4 et 6.6.3.

¹²⁴ Ici, le terme « cube » fait aussi bien référence à un cube (trois dimensions) qu'à un hypercube (n dimensions).

La mesure associée à un fait global est donc la mesure raster d'une mesure ROLAP. Elle peut résulter des quatre combinaisons possibles de faits détaillés/non-détaillés et raster/ROLAP tels que représentés à la Figure 70.

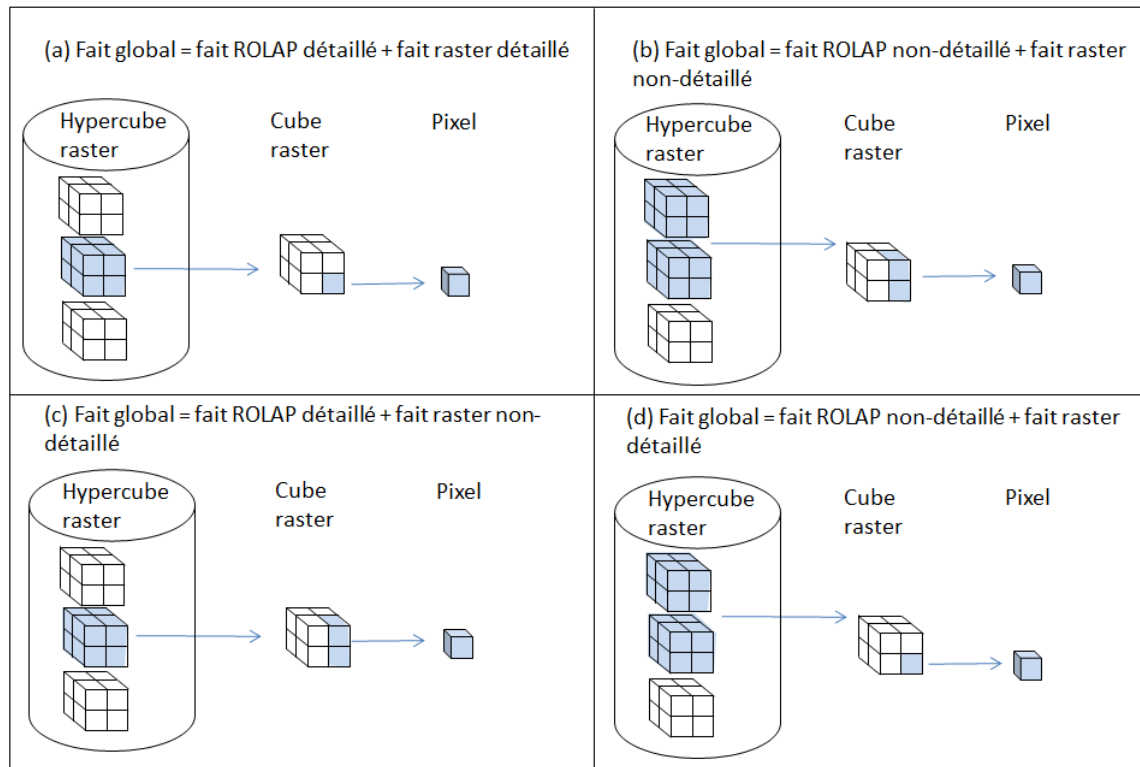


Figure 70. Différentes combinaisons de faits ROLAP et raster pour un fait global

6.3.2. Dimensions, membres, hiérarchies et cuboïdes

Rappelons que les membres d'une dimension raster sont identifiés par les index d'un cube raster et que les membres d'une dimension ROLAP sont identifiés par la clé primaire des enregistrements d'une table. Les propriétés hiérarchiques des dimensions sont donc très différentes entre le modèle raster et le modèle ROLAP. La gestion des hiérarchies des dimensions ROLAP est la même que dans tous les modèles d'entrepôts de données relationnels classiques (voir section 1.2.4). Par contre, la gestion des hiérarchies sur les dimensions raster est spécifique à notre modèle (voir sections 5.3.2, 5.4.1 et 5.4.2).

Nous avons déjà vu à la section 5.3 que les dimensions spatiales planimétriques X , Y étaient toujours modélisées par les dimensions raster respectives C et R . Grâce aux caractéristiques de la topologie raster, une hiérarchie commune à ces deux dimensions peut être définie implicitement à partir de facteurs d'échelles caractérisant les différents niveaux de cette hiérarchie. Nous avons ensuite considéré que la hiérarchie spatiale était toujours matérialisée par des cuboïdes raster (pyramides raster). Dans le modèle ROLAP, les cuboïdes raster matérialisant les différents niveaux de la hiérarchie spatiale sont considérés comme des mesures ROLAP distinctes. En effet, puisque seuls les faits raster (incluant les dimensions raster) sont affectés par les différents niveaux d'une hiérarchie raster (en l'occurrence, la hiérarchie spatiale), il est logique qu'ils partagent exactement les mêmes

faits ROLAP (incluant les dimensions ROLAP). Dans le modèle ROLAP, une opération de forage spatial (ou éventuellement de forage non-spatial sur une hiérarchie de B matérialisée) consiste donc à simplement changer de mesure ROLAP dans l'analyse (en d'autres mots : projeter¹²⁵ un attribut différent de la table des faits). Le schéma logique de la Figure 71 représente le schéma de la Figure 69 enrichi d'une hiérarchie spatiale à trois niveaux.

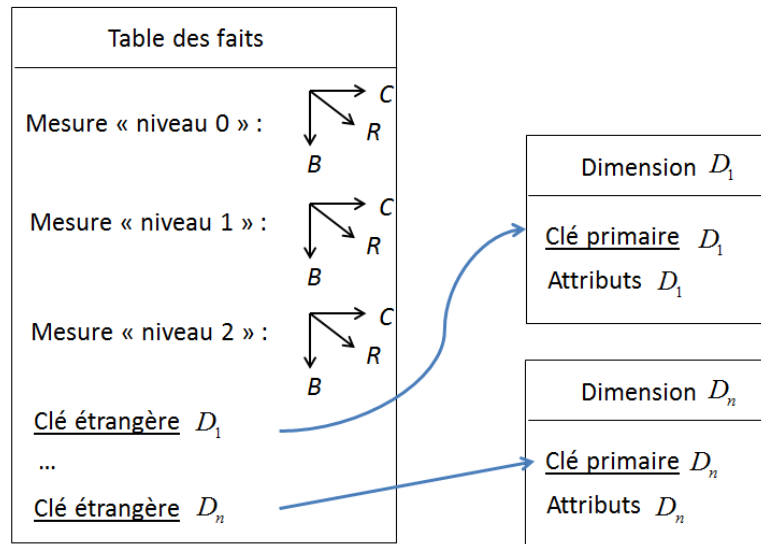


Figure 71. Schéma logique d'un hypercube raster comptant 3 niveaux spatiaux matérialisés par des cuboïdes

Si des cuboïdes sont matérialisés à la fois pour R , C et pour B , une mesure ROLAP distincte doit être associée à chaque combinaison possible de niveaux de la hiérarchie spatiale et de la hiérarchie de B (voir section 5.4.1), sauf dans le cas particulier où B modélise une dimension continue représentant la troisième dimension spatiale H (voir section 5.4.2.1).

Lorsque B présente une hiérarchie non-matérialisée, celle-ci peut être définie dans une¹²⁶ ou plusieurs¹²⁷ table(s) dimensionnelle(s) de manière quasi similaire à une dimension ROLAP. La seule différence réside dans le fait que la clé primaire des membres les plus détaillés ne les relie pas à la table des faits, mais directement à l'index raster de B . Dans le cas particulier où B modélise une dimension continue, nous avons vu à la section 5.4.2 qu'il était possible de lui attribuer une hiérarchie implicite basée sur la valeur des index de ses membres détaillés. La dimension B fera l'objet d'une analyse plus approfondie dans la section 6.8.

Finalement, comme dans tout modèle ROLAP, il est possible de matérialiser des cuboïdes sur les dimensions relationnelles (« hypercuboïde ROLAP »). Un hypercuboïde ROLAP d'une dimension relationnelle est une nouvelle table des faits qui est d'une part reliée à une table du niveau dimensionnel matérialisé (ou pas si le niveau est dégénéré), et

¹²⁵ La projection en algèbre relationnel est équivalente à la clause SELECT en SQL.

¹²⁶ Schéma en étoile.

¹²⁷ Schéma en flocon.

d'autre part reliée aux tables des autres dimensions relationnelles. Celles-ci sont donc partagées avec la table des faits de l'hypercube de base (schéma en galaxie ou en constellation). Comme nous l'avons vu à section 1.2.5, l'ensemble des tables des faits représentant une combinaison de niveaux ROLAP matérialisés forment un treillis d'hypercuboides dont le volume croît de manière exponentielle avec le nombre de niveaux matérialisés.

6.3.3. Membres spatialisés

Pour rappel (section 5.5), un membre spatialisé, ou membre géographique, est défini par un ensemble de membres spatiaux qui font partie à la fois de la dimension X et de la dimension Y . Nous avons vu qu'une couche de membres spatialisés pouvait être mise efficacement en relation avec un cube raster lorsqu'elle est modélisée par un raster dimensionnel. Un raster dimensionnel présente la propriété importante d'être intégrable à la volée au moment de l'analyse de l'utilisateur (raster dimensionnel intégré).

Puisque les relations entre un raster dimensionnel et cube raster sont établies via les dimensions R et C (ou Y , X), les membres spatialisés sont bien liés au modèle raster, et non au modèle ROLAP. Après son intégration, une couche de membres spatialisés intervient d'ailleurs uniquement dans les opérations élémentaires d'agrégation zonale et de filtrage spatial propres au modèle raster (voir section 5.7).

6.4. Métadonnées des mesures ROLAP

Nous avons vu qu'un fait ROLAP entraînait l'agrégation de cubes raster (mesures ROLAP) via une opération comparable à un « drill across ». Or, pour qu'une telle opération puisse être réalisée, les cubes de données impliqués doivent partager le même schéma multidimensionnel et les mêmes instances de dimension (voir section 1.3.2).

D'après notre définition de la notion d'hypercube raster (voir section 6.2), les cubes de données des mesures ROLAP partagent bien le même schéma multidimensionnel dans l'espace raster (R , C). Puisque notre modèle est un modèle SOLAP, les cubes de données doivent également partager le même schéma multidimensionnel dans l'espace géographique (X , Y). Nous avons vu à la section 5.2 que le passage de l'espace raster vers l'espace géographique était défini par les métadonnées spatiales d'un cube raster : SCR (système de coordonnées de référence de l'espace géographique), h_R, h_C (homothétie de l'espace raster), t_R, t_C (translation de l'espace raster). Tous les cubes raster d'une mesure ROLAP doivent donc partager les mêmes métadonnées¹²⁸ spatiales pour partager le même schéma multidimensionnel (X , Y).

Un raisonnement similaire est appliqué à la dimension B lorsque celle-ci modélise une dimension continue. Dans le cas d'une dimension B quelconque dont la hiérarchie est non-matérialisée, nous avons vu à la section 6.3.2 que tous les cubes associés à une même

¹²⁸ En réalité, si tous les paramètres de translation sont définis de telle sorte que les paramètres d'un premier cube (t_R, t_C) et les paramètres d'un second cube ($t_{R'}, t_{C'}$) satisfont la condition $Z_{R'}, Z_{C'} \in \mathbb{Z}$ (voir section 5.5.2), les cubes partagent bien le même schéma (X , Y), mais ne peuvent pas partager la même instance (x , y).

mesure ROLAP peuvent partager la même dimension par l'intermédiaire d'une table dimensionnelle.

Les métadonnées spatiales (et éventuellement les métadonnées spécifiques d'une dimension B continue) des mesures ROLAP atomiques permettent donc de fixer leur schéma multidimensionnel. Pour fixer la même instance sur tous les cubes raster (en d'autres mots : pour que les cubes partagent les mêmes membres dimensionnels), ils doivent partager exactement les mêmes métadonnées dimensionnelles : N_R, N_C, N_B .

Pour conséquent, un attribut « mesure cube raster¹²⁹ » (ou « mesure ROLAP ») de la table des faits doit être contraint aux mêmes métadonnées dimensionnelles (N_R, N_C, N_B) et aux mêmes métadonnées spatiales (SCR, t_R, t_C, h_R, h_C). Dans ce cas, les faits raster sont indexés exactement de la même manière dans tous les cubes et les agrégations sont donc physiquement optimisées (voir paragraphe suivant et Figure 73a). Ces contraintes peuvent être définies dans une table de métadonnées¹³⁰ où chaque enregistrement est relié à un attribut « mesure ROLAP » d'une table des faits (Figure 72). Si la dimension B est continue, la table des métadonnées contient aussi des contraintes sur les paramètres t_B, h_B et SCR de B .

Table des métadonnées	
Identification de la mesure	Table des faits
	Mesure ROLAP
Métadonnées spatiales	SCR
	Paramètres de translation (t_R, t_C)
	Paramètres d'homothétie (h_R, h_C)
Métadonnées dimensionnelles	N_R
	N_C
	N_B

Figure 72. Table des métadonnées d'un hypercube raster

Notons que si les métadonnées spatiales et dimensionnelles de R, C ne sont pas fixées dans la table des faits pour chaque attribut « mesure ROLAP », toutes les opérations d'agrégation sur ces mesures (« drill across ») nécessitent une homogénéisation de ces métadonnées par une série d'opérations similaires à celles présentées à la section 5.5.3. Dans le cas particulier où seules les métadonnées spatiales SCR, h_R et h_C sont partagées, l'espace de stockage peut être optimisé, mais pas les opérations d'agrégation (Figure 73b). Or, nous avons vu à la section 1.2 que dans un entrepôt de données, il était préférable d'optimiser les requêtes au détriment de l'espace de stockage (approche décisionnelle)

¹²⁹ Rappelons qu'un attribut « mesure ROLAP » (ou mesure « cube raster ») peut être un cube raster de base ou un cuboïde raster matérialisant un niveau de la hiérarchie spatiale (ou éventuellement un niveau de la hiérarchie spatiale et un niveau de la hiérarchie de B).

¹³⁰ Dans PostGIS raster, cette table des métadonnées correspond à la vue « raster_column ».

plutôt que l'inverse (approche transactionnelle). Nous rejetons donc cet aspect de notre modélisation. Un raisonnement similaire est applicable aux métadonnées de la dimension B si celle-ci est continue. Si B est quelconque, il est impossible d'homogénéiser les métadonnées dimensionnelles N_B au moment de la requête (absence d'un système de référence absolu comparable au SCR sur cette dimension).

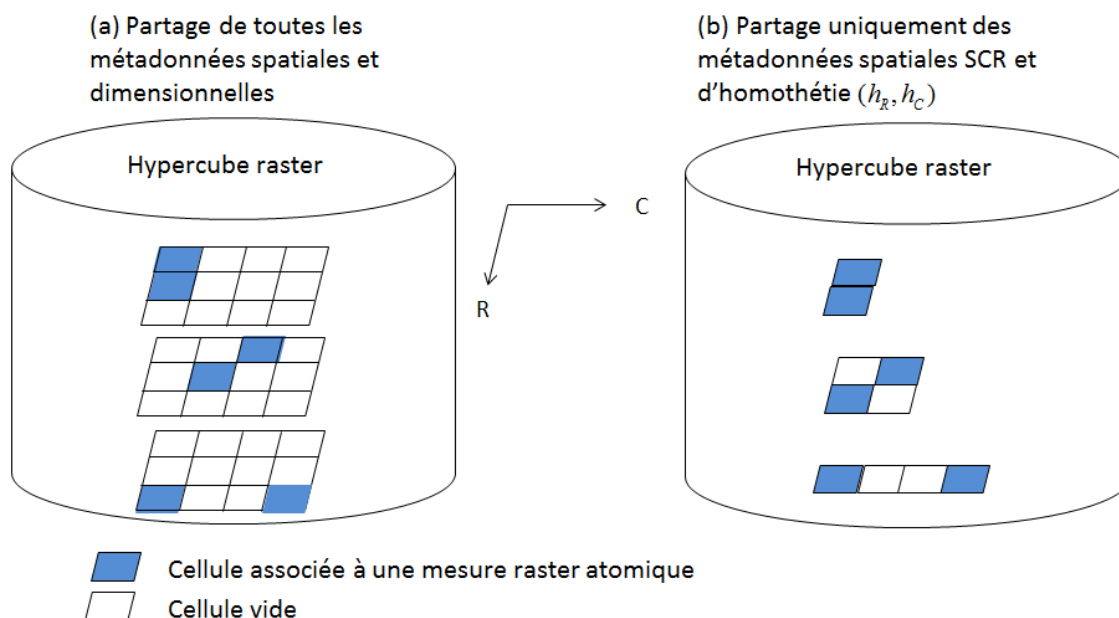


Figure 73. Illustration de l'espace de stockage occupé par un hypercube raster avec partage total (a) et partiel (b) des métadonnées d'une mesure ROLAP (pas de dimension B dans cet exemple)

6.5. Exemple d'hypercube raster

Dans les prochaines sections, portant sur la définition des opérations SOLAP raster dans notre modèle ROLAP, nous nous baserons toujours sur le même exemple d'hypercube raster qui est une extension du cube raster présenté à la section 5.6.

Le schéma conceptuel de cet hypercube (Figure 74) se différencie de celui de la section 5.6 par l'ajout d'une dimension ROLAP : les types de délits. Par conséquent, la mesure est également modifiée en « nombre de délits ». Pour retrouver le nombre de cambriolages, il suffit d'appliquer une coupe sur le membre « cambriolage » de la dimension ROLAP « type de délit » (voir section 6.7). Dans cet exemple, la dimension ROLAP n'est pas hiérarchisée car nous avons vu que la gestion d'une hiérarchie ROLAP dans notre modèle n'est pas différente de celle d'un modèle ROLAP classique (schéma en étoile, flocon, etc.). Des exemples de dimensions ROLAP hiérarchisées seront illustrés dans le chapitre 8.

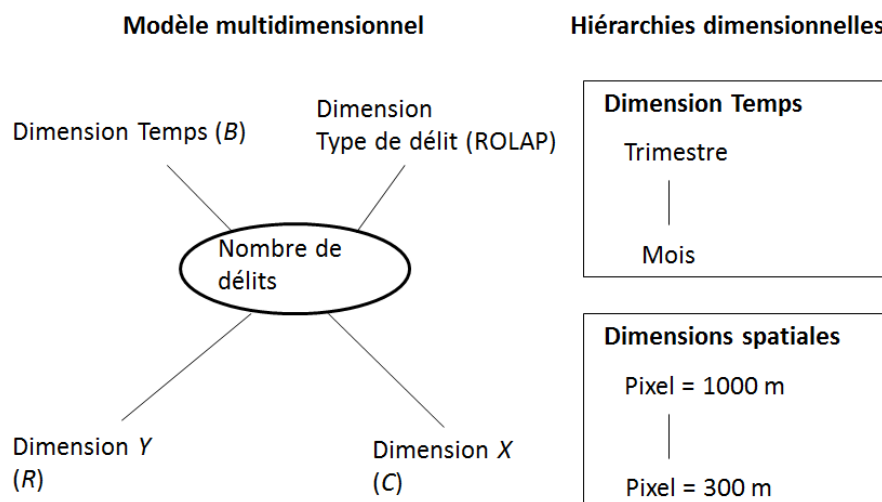


Figure 74. Exemple de schéma conceptuel d'un hypercube raster

Notons que dans l'instance de ce schéma, la dimension temporelle compte 12 membres détaillés (les mois de 2012) et la dimension « type de délit » compte 11 membres détaillés.

6.6. Opérations ROLAP élémentaires

Dans la section 5.7, nous avons défini une chaîne d'opérations élémentaires appliquées sur un cube raster (modèle SOLAP raster) pour obtenir le résultat affiché par une interface. La présente section a pour but de décrire cette chaîne dans le modèle ROLAP (donc appliquée sur un hypercube raster). Le rôle de la chaîne ROLAP est donc toujours le même que la chaîne raster : la réduction de la dimensionnalité de l'hypercube raster à chaque opération jusqu'à obtenir la forme souhaitée par l'interface utilisateur.

Puisque le modèle ROLAP manipule des cubes de données issus du modèle raster (mesure ROLAP), la chaîne du modèle ROLAP est composée à la fois d'opérations raster et d'opérations ROLAP. Elle contient toujours deux grands types de traitements, les filtrages et les agrégations, et ceux-ci sont appliqués dans un certain ordre :

- **Filtrage vertical ROLAP**
- Filtrage vertical raster
- **Filtrage horizontal ROLAP**
- Filtrage horizontal raster
- **Agrégation partielle ROLAP**
- Agrégation partielle raster
- **Agrégation totale ROLAP**
- Agrégation totale raster

La chaîne d'opérations élémentaires du modèle ROLAP dans l'ordre présenté ici est schématisée dans la Figure 75. Les opérations raster sont indiquées en rouge et les opérations ROLAP sont indiquées en bleu. Nous verrons à la section 6.6.4 que l'ordre relatif dans lequel sont exécutées les opérations élémentaires d'agrégation peut être

différent afin de privilégier certaines opérations SOLAP au détriment d'autres. D'ici là, nous nous baserons sur l'hypothèse d'ordre présenté par la Figure 75 (et dans la liste ci-dessus). Les opérations raster étant pratiquement les mêmes que celles de la section 5.7, nous allons les passer rapidement en revue dans le paragraphe suivant. Les nouvelles opérations ROLAP, quant à elles, seront décrites dans prochaines sections.

Le filtrage vertical raster consiste en la sélection de la mesure ROLAP (attribut de la table des faits) correspondant à la combinaison de niveaux hiérarchiques matérialisés souhaitée par l'utilisateur (par exemple, le niveau spatial « pixel = 1000 m »). De cette sélection résulte l'hypercube ROLAP pré-calculé sur lequel la suite de la chaîne d'opérations élémentaires sera poursuivie. Les autres opérations raster sont les mêmes que celles présentées à la section 5.7, mis à part qu'elles sont appliquées sur un hypercube raster et non plus sur un cube raster. Rappelons qu'un hypercube raster est un ensemble de cubes¹³¹ raster partageant les mêmes métadonnées spatiales et dimensionnelles. Ces opérations raster sont donc appliquées de la même manière sur tous les cubes raster (mesure ROLAP) qui composent l'hypercube. Par exemple, dans un filtrage horizontal spatial, le nombre de membres spatiaux ($N_{R,C}$) est réduit à f (nombre de filtres) pour toutes les mesures ROLAP de l'hypercube raster traité. Le résultat est donc logiquement un sous-hypercube. De la même manière, une agrégation (partielle ou totale) raster est appliquée de la même manière sur tous les cubes composant l'hypercube traité et le résultat est donc logiquement un hypercube. Notons que dans ce contexte, le terme « hypercube » est assez large puisqu'en fonction de l'opération SOLAP appliquée, un hypercube peut être composé de plusieurs ou d'un seul cube (agrégation totale de toutes les dimensions ROLAP), et les cubes composant l'hypercube peuvent être réduits à deux dimensions (raster résultant de l'agrégation totale de B), une dimension (liste de valeurs selon B résultant de l'agrégation totale de R et C) ou zéro dimension (valeur unique résultant d'une agrégation totale de R, C, B).

Vu que les dimensions spatiales (X, Y) sont toujours gérées par les dimensions raster C et R , les opérations spatiales, y compris celles impliquant des membres spatialisés (voir section 5.7), sont toujours gérées dans le modèle raster et non dans le modèle ROLAP. Par conséquent, les dimensions ROLAP et les opérations élémentaires ROLAP sont toujours non-spatiales. Lors des prochaines sections, nous allons décrire les opérations élémentaires ROLAP dans leur ordre d'apparition dans la chaîne proposée, comme nous l'avons fait pour les opérations élémentaires raster à la section 5.7.

¹³¹ Dans certains cas, les cubes qui composent l'hypercube peuvent être réduits à deux (raster), une (liste de valeurs), voire zéro (valeur unique) dimension(s).

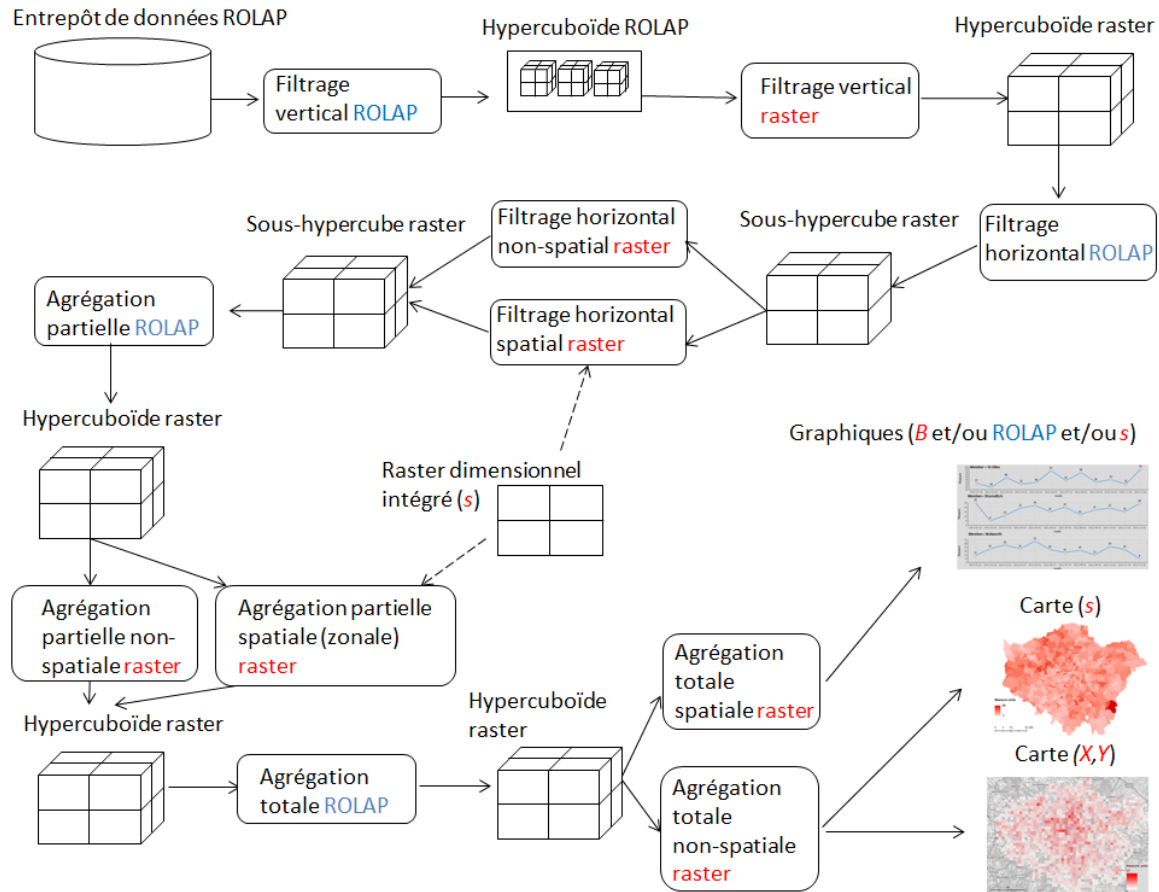


Figure 75. Chaîne d'opérations élémentaires appliquée sur un entrepôt de données raster relationnel

6.6.1. Filtrage vertical ROLAP

Le filtrage vertical ROLAP est la première maille de la chaîne d'opérations élémentaires ROLAP. Il permet à l'utilisateur de choisir les niveaux dimensionnels ROLAP dont la hiérarchie est matérialisée (voir section 6.3.2). Cette opération élémentaire est donc responsable du forage sur ces dimensions (appliquer un forage sur une hiérarchie ROLAP matérialisée implique donc de recommencer toute la chaîne de traitements). Il est opéré par sélection de la table des faits adéquate hors de l'entrepôt de données ROLAP, dans lequel il existe une table des faits pour chaque combinaison possible de niveau de ces dimensions ROLAP particulières. Par exemple, si les dimensions « temps » et « type de délit » étaient toutes les deux des dimensions ROLAP¹³² à hiérarchie matérialisée, le résultat d'un filtrage vertical ROLAP pourrait être une sélection de la table des faits (hypercuboïde ROLAP) correspondant aux cambriolages du mois de septembre 2012.

Puisque la matérialisation des hypercuboïdes ROLAP dans notre architecture présente peu d'originalité par rapport à ce qui se fait couramment dans une architecture ROLAP classique telle que GeoMondrian (Pentaho, 2014), nous ne développerons pas cet

¹³² Dans l'exemple d'hypercube de la section 6.5, le temps est modélisé par la dimension *B* et les types de délits sont modélisés par une dimension ROLAP.

aspect davantage. Dans la suite de cette thèse, toutes les dimensions ROLAP traitées présenteront toujours une hiérarchie non-matérialisée et par conséquent, l'ED ROLAP sera uniquement composé de l'hypercube ROLAP de base. Le filtrage vertical ROLAP est donc limité à cette description (contrairement au filtrage vertical raster).

6.6.2. Filtrage horizontal ROLAP

Un filtrage horizontal sur un membre ROLAP (filtre ROLAP) d'un hypercube raster est une sélection des enregistrements de la table des faits (faits ROLAP détaillés) en relation avec ce membre ROLAP. Lorsque le filtrage d'une dimension ROLAP comprend plusieurs filtres, ceux-ci sont combinés par un « ou » logique. Par exemple :

membre « mois » = « janvier » OU membre « mois » = « février »

Lorsque plusieurs filtres sont appliqués (un par dimension ROLAP), ceux-ci sont combinés par un « et » logique. Par exemple :

(membre « mois » = « janvier » OU membre « mois » = « février ») ET (membre « type de délit » = « cambriolage »)

Le résultat de cette opération sur un hypercube raster est donc un sous-hypercube raster (réduction du nombre de membres ROLAP). Le filtrage horizontal ROLAP est responsable de l'opération de coupe sur les dimensions ROLAP. Par conséquent, une nouvelle coupe appliquée sur ces dimensions implique la reprise de la chaîne d'opérations élémentaires à cette étape.

Notons qu'en SQL, les filtres horizontaux ROLAP sont indiqués dans la clause « WHERE ».

6.6.3. Agrégations ROLAP

Comme nous l'avons déjà mentionné, les opérations d'agrégations ROLAP (partielles ou totales) sont similaires à des opérations « drill across » entre plusieurs cubes de données (en l'occurrence, des cubes raster). Une agrégation ROLAP ne peut être appliquée que sur un ensemble de mesures atomiques ROLAP appartenant au même attribut « mesure ROLAP ». Puisque tous les cubes raster d'un même attribut « mesure ROLAP » partagent les mêmes métadonnées spatiales et dimensionnelles (voir section 6.4), le résultat de l'agrégation ROLAP de n mesures ROLAP est un cube raster associé à des métadonnées identiques à celles des mesures ROLAP agrégées. Si la fonction d'agrégation employée est la somme, chaque fait raster détaillé du cube agrégé est associé à une mesure raster calculée de la manière suivante :

$$p_{b,r,c} = p_{b,r,c}(0) + p_{b,r,c}(1) + \dots + p_{b,r,c}(n-1)$$

Dans cette formule, $p_{b,r,c}$ est la mesure raster associée au fait raster de coordonnées ($R=r$, $B=b$, $C=c$) dans le cube raster résultant de l'agrégation. Les mesures ROLAP à agréger sont ordonnées et numérotées de 0 à $n-1$ ($n \in \mathbb{N}$).

La Figure 76a montre qu'en l'absence de dimension *B*, cette opération correspond à une opération locale d'algèbre de carte. Si une dimension *B* est présente, la Figure 76b montre qu'il s'agit d'une opération locale d'algèbre de carte cubique (Mennis *et al*, 2005).

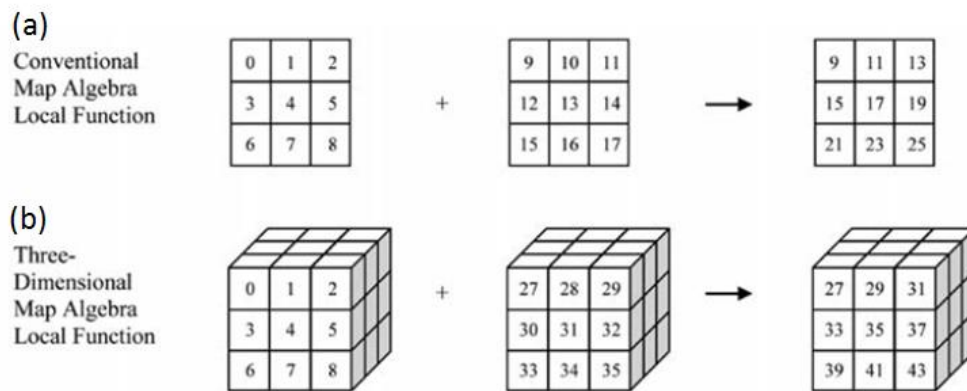


Figure 76. Opération locale d'algèbre de carte conventionnel (a) et cubique (b) (Mennis *et al*, 2005)

En SQL, l'agrégation est indiquée sur la mesure ROLAP projetée dans la clause « SELECT ».

6.6.3.1. Agrégation partielle ROLAP

L'agrégation partielle ROLAP est appliquée sur les dimensions ROLAP dont la hiérarchie n'est pas matérialisée. Elle dépend du niveau hiérarchique de la dimension ROLAP représentée dans l'interface et elle n'a de raison d'être que si ce niveau est non-détaillé.

Le résultat de l'agrégation partielle selon le niveau *L* d'une dimension ROLAP est un hypercuboïde raster calculé à la volée où le nombre de membres de la dimension ROLAP est réduit au nombre de membres du niveau *L* (les opérations élémentaires suivantes de la chaîne sont donc appliquées sur l'hypercuboïde résultant). Dans cet hypercuboïde, une mesure d'un fait ROLAP (qui, rappelons-le, est un cube raster) est calculée par agrégation ROLAP¹³³ de toutes les mesures des faits détaillés (niveau 0) en relation hiérarchique avec le fait ROLAP de niveau *L*. Si la dimension temporelle de l'exemple de la Figure 74 était une dimension ROLAP dégénérée, un exemple d'agrégation partielle sur le membre 1 du niveau « trimestre » pourrait être exprimé en SQL de la manière suivante (en admettant que la fonction « SUM » agrège les cubes raster de la manière expliquée à la section précédente) :

```
SELECT trimestre, SUM( "mesure_1000m" )
FROM cube_raster
GROUP BY trimestre
```

¹³³ Voir section précédente.

L'agrégation partielle ROLAP est à la base de l'opération de forage sur une dimension ROLAP dont la hiérarchie n'est pas matérialisée. La chaîne de traitements doit donc reprendre à cette étape lorsque l'utilisateur souhaite forer une telle dimension.

6.6.3.2. Agrégation totale ROLAP

L'opération élémentaire d'agrégation totale ROLAP réduit à un seul membre (« all ») toutes les dimensions ROLAP de l'hypercube raster qui ne sont pas affichées par l'interface. Toutes les mesures de l'hypercube raster sont ainsi totalement agrégées selon ces dimensions. Le résultat est un hypercuboïde raster (ou éventuellement un cuboïde si moins de quatre dimensions¹³⁴ sont représentées dans le résultat).

Dans une requête SQL de type « GROUP BY », cette opération élémentaire est effectuée de manière implicite. Ainsi, dans la requête de la section précédente, toutes les autres dimensions ROLAP éventuellement présentes dans le cube (par exemple : les types de délits) sont totalement agrégées. Il est cependant important de bien séparer les agrégations ROLAP partielles et totales dans la chaîne de traitement (voir Figure 75) car celles-ci peuvent éventuellement nécessiter des fonctions d'agrégation différentes (distributives, algébriques ou holistiques) dans certains cas particuliers. Nous reviendrons sur ce point à la section 8.4.

6.6.4. Ordre des opérations élémentaires

Nous venons de décrire les différentes étapes de la chaîne d'opérations élémentaires ROLAP selon un ordre précis afin d'expliquer clairement le rôle de ces différentes opérations. Comme nous allons le voir, il est possible d'appliquer les opérations élémentaires dans un ordre différent afin d'optimiser certains traitements au détriment d'autres. Le résultat final sera toujours le même, peu importe l'ordre, pour autant que la fonction d'agrégation utilisée à toutes les opérations soit identique, et distributive ou algébrique.

Dans tous les cas, le début de la chaîne (chaîne de filtrage) doit toujours être le même que celui proposé :

- Filtrage vertical ROLAP
- Filtrage vertical raster
- Filtrage horizontal ROLAP
- Filtrage horizontal raster

Il est évident que la chaîne doit commencer par les filtres verticaux dans l'ordre ROLAP puis raster : la table des faits (filtrage ROLAP) doit être sélectionnée avant de pouvoir sélectionner l'attribut « mesure ROLAP » adéquat (filtrage raster). En ce qui concerne les filtres horizontaux, l'ordre raster puis ROLAP impliquerait un découpage inutile des cubes raster (mesures ROLAP) qui ne sont pas repris par le filtrage ROLAP.

¹³⁴ Y compris les dimensions raster *R*, *C*, *B*.

Les agrégations, quant à elles, peuvent être appliquées dans n'importe quel ordre afin d'optimiser les opérations SOLAP de forage ou de changement d'interface sur les dimensions raster ou les dimensions ROLAP. En résumé, plus un traitement est situé loin dans la chaîne et plus l'opération SOLAP dont il est responsable est optimisée lorsqu'elle est appelée de manière consécutive. En effet, une opération en fin de chaîne est appliquée sur un hypercube simplifié par les opérations précédentes. Reprenons comme exemple l'ordre proposé dans la section 6.6 :

- Agrégation partielle ROLAP
- Agrégation partielle raster
- Agrégation totale ROLAP
- Agrégation totale raster

Dans ce cas, les changements de dimensions raster affichées dans l'interface¹³⁵ (carte ou graphique) sont les opérations SOLAP les plus avantageées, suivis des changements de dimensions ROLAP affichées dans l'interface, suivis des forages consécutifs sur une hiérarchie *B* non-matérialisée ou sur des membres spatialisés, puis des forages consécutifs sur les hiérarchies ROLAP non-matérialisées. Un autre exemple serait d'appliquer les opérations selon cet ordre :

- Agrégation partielle raster
- Agrégation totale raster
- Agrégation totale ROLAP
- Agrégation partielle ROLAP

Dans ce cas, les forages consécutifs sur une hiérarchie ROLAP non-matérialisée sont les plus avantageés, suivis des changements de dimensions ROLAP affichés dans l'interface, suivis des changements de dimensions raster affichés par l'interface, suivis des forages sur une dimension *B* non-matérialisée ou sur des membres spatialisés. Toutes les combinaisons d'ordre possibles parmi ces quatre opérations élémentaires d'agrégation sont donc envisageables, mais il s'avère difficile d'en établir une plus globalement plus avantageux que les autres. Cela dépend avant tout des données, de leur modélisation et de l'analyse de l'utilisateur.

6.7. Opérations SOLAP et interfaces du modèle ROLAP

Du point de vue de l'utilisateur, les opérations SOLAP dans le modèle ROLAP sont exactement les mêmes que celles présentées dans les sections 5.8 et 5.9 pour le modèle purement raster, sachant que les opérations sur les dimensions ROLAP sont traitées comme des opérations non-spatiales. Les coupes peuvent ainsi être combinées sur plusieurs dimensions, qu'elles soient ROLAP ou raster (non-spatiale et spatiale) et un forage peut avoir lieu sur n'importe quel type de dimension. Il en va de même pour les comparaisons.

Puisque notre modèle ROLAP est capable de gérer plusieurs dimensions non-spatiales, il est maintenant possible d'exploiter les interfaces tabulaires ou graphiques

¹³⁵ Par exemple, affichage d'une carte représentant *R,C* puis d'un graphique représentant *B*.

bidimensionnelles pour représenter plusieurs de ces dimensions (ROLAP ou B). La Figure 77 illustre le résultat d'une opération SOLAP affichant deux dimensions non-spatiales dans l'interface graphique et les deux dimensions spatiales dans l'interface raster. Ce résultat provient d'une coupe spatiale isolant le quartier de Notting Hill et d'une coupe non-spatiale sur la dimension ROLAP « type de délit » isolant les rackets (« robbery »), les actes de vandalisme et incendies criminels (« criminal damage and arson »), les cambriolages (« burglary ») et les vols à l'étalage (« shoplifting »). Puisque la carte n'affiche que les dimensions spatiales, sa mesure est agrégée pour les quatre types de délits. Puisque les graphiques n'affichent pas les dimensions spatiales, leurs mesures sont agrégées pour tout le quartier de Notting Hill.

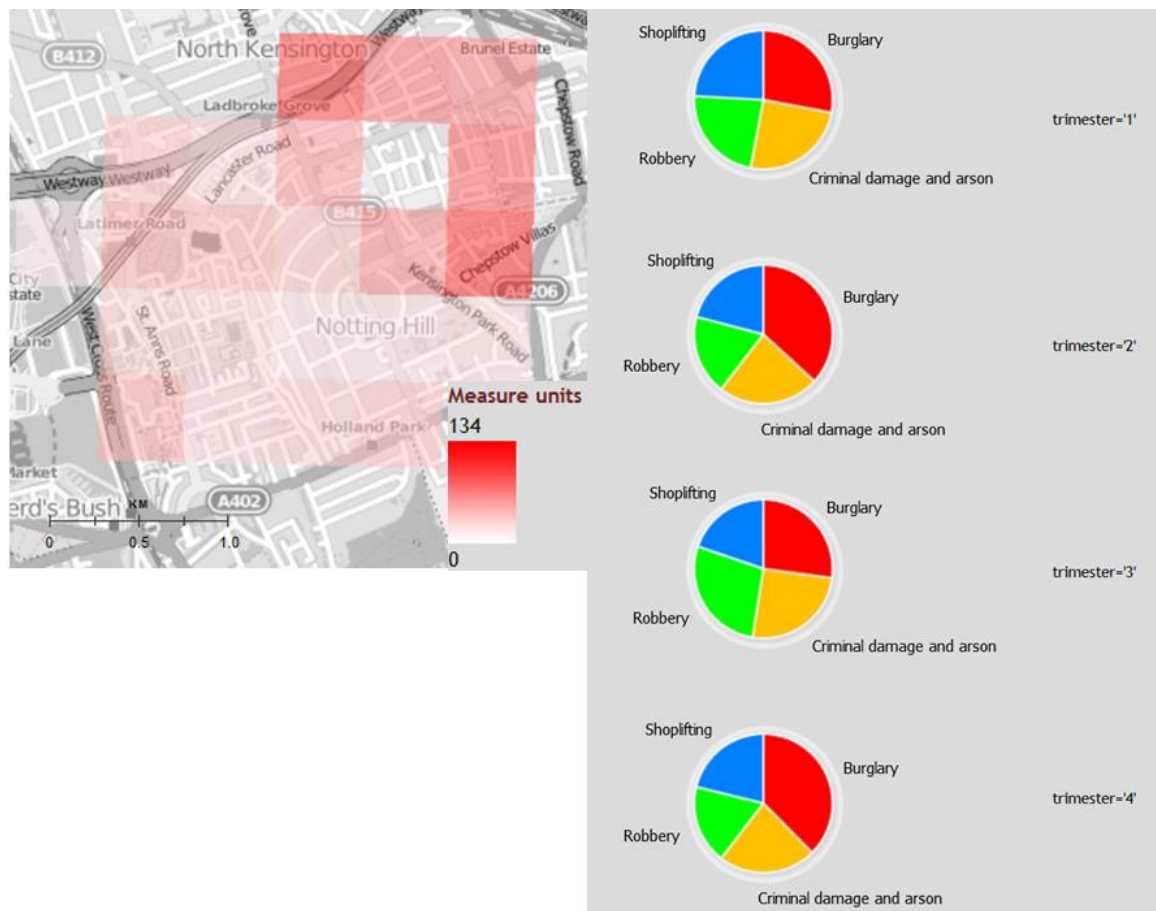


Figure 77. Interface graphique affichant une dimension B (type de délit : 4 membres) et une dimension ROLAP (trimestre : 4 membres)

Pour obtenir le résultat de la Figure 77, la chaîne d'opérations élémentaires a été appliquée de la manière suivante (dans l'ordre de la Figure 75) :

- Filtrage vertical ROLAP : aucun
- Filtrage vertical raster : mesure ROLAP « pixel = 300 m »
- Filtrage horizontal ROLAP : membres « burglary », « criminal damage and arson », « robbery » et « shoplifting »
- Filtrage horizontal raster :
 - o Spatial : raster dimensionnel comportant un membre « Notting Hill »
- Agrégation partielle ROLAP : aucune
- Agrégation partielle raster :
 - o Non-spatiale : niveau « trimestre »
- Agrégation totale ROLAP : dimension « type de délit » (carte)
- Agrégation totale raster :
 - o Non-spatiale (carte)
 - o Spatiale (graphiques)

6.8. Intérêt de la dimension *B* dans le modèle ROLAP

Après avoir décrit les caractéristiques de notre modèle raster relationnel (ROLAP), quelques éclaircissements s'imposent par rapport à l'intérêt de la dimension raster *B* dans ce modèle. A plusieurs reprises, nous avons pu remarquer qu'une dimension *B* quelconque présentait d'importantes similarités avec une dimension ROLAP :

- Elles sont toutes les deux non-spatiales.
- Leurs hiérarchies respectives sont toutes les deux gérées de manière purement relationnelle (en l'absence de cuboïde matérialisé).

Alors que cette dimension était justifiée dans un modèle purement raster (voir chapitre 5) puisqu'elle était la seule à pouvoir gérer une dimension non-spatiale, sa présence semble déranger dans le modèle ROLAP. En effet, les dimensions ROLAP remplissent le même rôle que *B* en s'appuyant davantage sur la capacité de stockage des SGBD relationnels en termes de nombres d'enregistrements d'une table (qui est plus solide que celle des tableaux multidimensionnels¹³⁶). De plus, nous avons vu qu'une dimension ROLAP ne présentait pas de problème de faible densité, contrairement à une dimension *B*. Dans un cas quelconque, il semble donc préférable de se passer de la dimension *B* en implémentant un SOLAP raster uniquement sur les deux dimensions raster *R*, *C* et les dimensions ROLAP.

Cependant, nous avons vu à la section 5.4.2 que la dimension *B* pouvait présenter des avantages similaires aux dimensions *R* et *C* lorsqu'elle modélise une dimension continue. Grâce à un jeu de métadonnées permettant le passage de l'espace raster vers l'espace modélisé par *B* (homothétie, translation et système de référence), il devient possible de faire le rapprochement entre la topologie raster sur l'axe *B* et la proximité des membres de *B* dans son système de référence. Une hiérarchie sur *B* peut ainsi être modélisée de manière implicite et une sélection des membres de *B* peut être effectuée sur base d'opérations logiques dans le système de référence de la dimension modélisée par *B*

¹³⁶ A ce jour, PostGIS raster est d'ailleurs assez limité par la taille des raster qu'il permet de stocker. D'après notre expérience, environ un million de pixels codés sur 32 bits semblent être la limite de stockage acceptée par le SGBD.

(par exemple : sélection des membres temporels compris entre le 21 février 2015 à 3h43 et le 25 février 2015 à 18h57). Ce genre d'opération exploite ainsi des membres indexés de manière cohérente par rapport à la dimension modélisée et est donc plus efficace que dans une modélisation ROLAP. La présence d'une dimension raster B est donc particulièrement justifiée dans le modèle raster relationnel quand elle modélise une dimension continue (par exemple : le temps ou la hauteur).

Finalement, comme nous le confirmera l'étude comparative de la section 8.5, précisons que les agrégations totales sont beaucoup plus efficaces sur les dimensions raster (assimilables à du MOLAP) que sur les dimensions ROLAP. Une dimension B , qu'elle soit continue ou non, bénéficie donc toujours des performances du MOLAP qui sont supérieures au relationnel pur lorsque la dimension est totalement agrégée (non affichée dans l'interface).

6.9. Conclusion

Le modèle présenté dans ce chapitre est l'extension relationnelle du modèle raster multidimensionnel présenté lors du chapitre précédent. L'idée est de considérer un cube raster comme une mesure d'un modèle ROLAP classique (mesure ROLAP). Les trois dimensions du cube raster (R , C , B) sont ainsi enrichies de dimensions supplémentaires gérées par le modèle ROLAP : les dimensions ROLAP. L'instance d'un tel schéma est donc un hypercube raster. Puisqu'un hypercube raster est composé à la fois de dimensions ROLAP et de dimensions raster (assimilables à du MOLAP), chaque élément de sa structure multidimensionnelle est défini à la fois dans le modèle raster et dans le modèle relationnel : hiérarchies raster et hiérarchies ROLAP, faits raster et faits ROLAP, mesures raster et mesures ROLAP, mais aussi cuboïdes raster et cuboïdes ROLAP. Puisque les mesures ROLAP sont des cubes raster, celles-ci sont agrégées par une opération de type « drill across » consistant à agréger entre elles les mesures de plusieurs cubes de données partageant les mêmes dimensions raster et la même instance. Pour que ces cubes partagent les mêmes dimensions spatiales X , Y et les mêmes instances spatiales x , y , ils doivent nécessairement être définis par les mêmes métadonnées spatiales (SCR , h_R, h_C, t_R, t_C) et dimensionnelles¹³⁷ (N_R, N_C). Afin de ne pas alourdir les opérations SOLAP par des transformations raster à la volée, ces métadonnées sont définies comme des contraintes sur les mesures ROLAP dans une table spécifique : la table de métadonnées. Une chaîne d'opérations élémentaires (filtrages et agrégations) permettant le calcul des mesures affichées par l'interface SOLAP intègre les opérations spécifiques aux dimensions ROLAP, et les opérations spécifiques aux dimensions raster doivent être appliquées sur un hypercube (et non plus sur un cube). Les interfaces et opérations SOLAP dans le modèle relationnel sont sensiblement les mêmes que dans le modèle raster, mis à part que les graphiques sont capables d'afficher plusieurs dimensions non-spatiales dans le modèle relationnel (le modèle purement raster ne pouvait pas manipuler plus d'une dimension non-spatiale). Tout comme le modèle purement raster, le modèle ROLAP est capable d'intégrer des membres spatialisés à la volée.

¹³⁷ Un raisonnement similaire est valable pour une dimension B continue.

Grâce cette extension relationnelle, notre modèle SOLAP raster peut gérer plusieurs dimensions non-spatiales en plus des deux dimensions spatiales X et Y . De plus, il est maintenant facile de l'implémenter dans un SGBD intégrant les données raster (par exemple : PostGIS ou Oracle) puisque les opérations élémentaires d'agrégation et filtrage raster correspondent respectivement à de l'algèbre de carte et à des découpages de grilles numériques géoréférencées bien connus du domaine raster.

Chapitre 7 : modèle raster continu

7.1. Introduction : hypercube raster continu

Comme nous l'avons vu à la section 2.1, un des avantages du format raster est sa facilité à modéliser des phénomènes spatialement continus. Un tel phénomène est alors stocké sous forme de champ (grid raster) où l'espace est divisé en mailles (pixels) associées à une valeur, et où la valeur d'un point quelconque du champ est assimilée à la valeur de la maille dans laquelle il est inclus. Puisque le modèle SOLAP présenté lors des deux chapitres précédents est uniquement composé de raster¹³⁸, il est capable de stocker des champs continus. Dans ce cas, une instance de notre modèle raster est appelée « hypercube¹³⁹ raster continu ».

Chaque cellule composant un hypercube raster continu est associée à une mesure atomique caractérisant la valeur d'un champ dans une maille définie par les dimensions spatiales X , Y (membre spatial), mais aussi par les dimensions non-spatiales B et ROLAP (membres non-spatiaux). Lorsque l'interface cartographique affiche les dimensions spatiales X , Y , elle affiche un champ résultant d'une agrégation non-spatiale. Dans tous les autres cas (interface cartographique affichant des membres spatialisés s ou interface graphique), l'interface affiche les résultats d'agrégations spatiales appliquées sur des champs, et définies par les dimensions non-spatiales (ou spatialisées) affichées.

Par rapport aux travaux présentés à la section 4.2, notre SOLAP continu se place dans la catégorie des modèles maillés¹⁴⁰ où l'interpolation des champs est implicite. En effet, tous les modèles exploitant des champs vectoriels (Ahmed & Miquel, 2005; Vaisman & Zimányi, 2009; Bimonte & Kang, 2010; Zaamoune *et al*, 2013) utilisent des opérations élémentaires d'interpolation combinées aux agrégations (sur les points d'échantillonnage) pour calculer la mesure attribuée à un fait spatial quelconque¹⁴¹. Dans les modèles vectoriels maillés (Mc Hugh, 2008; Gomez *et al*, 2010; Plante, 2014), l'attribution des valeurs des mailles par interpolation¹⁴² est effectuée en amont de l'entrepôt de données, lors de la phase d'intégration (voir chapitre 1). Puisqu'un fait spatial quelconque est toujours attribué à au moins une maille, la mesure associée à ce fait est donc toujours calculée par des opérations d'agrégation sur des mesures atomiques résultant d'une interpolation préalable. Le modèle développé dans la suite de ce chapitre repose donc sur un hypercube agréant des champs raster pré-calculés.

¹³⁸ Un hypercube raster (dimensions ROLAP et dimensions raster) est un ensemble de cubes raster (dimensions raster R , C , B) et un cube raster est un ensemble de raster (dimensions raster R , C).

¹³⁹ Ou « cube raster continu » si seulement trois dimensions sont modélisées.

¹⁴⁰ Dans cette catégorie, notre modèle se distingue des autres par son exploitation du format raster.

¹⁴¹ Nous appelons « fait spatial quelconque » n'importe quel fait attribué à un point quelconque de la couverture spatiale modélisée dans le cube de données.

¹⁴² Les valeurs des mailles sont calculées par des méthodes d'interpolation quand il s'agit d'un champ géographique. A l'inverse, les valeurs des mailles des images géographiques proviennent de capteurs (Donnay, 2005a).

La prochaine section sera introduite par une brève discussion sur les différentes méthodes d'interpolation utilisables lors de l'intégration des champs pré-calculés. Nous y argumenterons ensuite le choix d'une méthode particulière avec laquelle sera illustré notre modèle : l'estimation de la densité par la méthode des noyaux (« Kernel Density Estimation » ou KDE). Celle-ci présente l'intérêt particulier d'offrir une vision spatialement continue de données discrètes, jusqu'à présent absente des modèles proposés dans la littérature SOLAP. Dans la section 7.3, cette dernière méthode sera décrite d'un point de vue théorique et applicatif dans le domaine de la cartographie criminelle, puisque cette discipline est liée à notre second objectif de recherche (développement d'un SOLAP adapté aux besoins de la police). La section 7.4 nous permettra de définir et de démontrer les conditions nécessaires à l'intégration de champs KDE dans un hypercube raster continu, que nous appelons alors « hypercube raster KDE ». A partir d'un exemple d'hypercube raster KDE présenté à la section 7.5 (toujours sur des données de criminalité), plusieurs opérations orientées utilisateur seront illustrées afin de prouver l'utilité du SOLAP raster continu faisant l'objet de ce chapitre (section 7.6). Une conclusion sera ensuite dressée par rapport à ce qui aura été vu précédemment (section 7.7).

7.2. Interpolation des champs

Il existe plusieurs méthodes d'interpolation dont le résultat peut être stocké dans un champ raster (et donc dans un hypercube raster) : pondération inverse à la distance (Watson & Philip, 1985), krigeage (Davis, 1986; Oliver & Webster, 2010), voisins naturels (Sibson, 1981), splines avec ou sans interruption (Franke, 1982), surface de tendance (ESRI, 2012), *etc.* Toutes ces techniques se basent sur une hypothèse commune : les données présentent une autocorrélation spatiale (voir section 2.1.1). Le choix d'une méthode adéquate pour un jeu de données n'est cependant pas toujours évident. Certaines sont typiquement utilisées dans des domaines d'application spécifiques. Certaines font ressortir les tendances locales alors que d'autres font ressortir les tendances globales. Certaines sont plus adaptées à des phénomènes jugés aléatoires alors que d'autres sont plus adaptées à des phénomènes explicables (Mitas & Mitasova, 1999). Le choix de la méthode à utiliser dans un hypercube raster est donc très dépendant des caractéristiques des données interpolées et des échelles d'analyse exploitées par l'utilisateur. Il ne serait pas exclu d'utiliser des techniques différentes pour les différents cuboïdes matérialisant les niveaux d'échelle de la hiérarchie spatiale.

L'influence des différentes méthodes d'interpolation et de leurs paramètres sur les résultats des agrégations dans un hypercube raster constituerait un sujet d'étude intéressant, mais qui dépasse les objectifs fixés dans le cadre de cette thèse. Puisque notre but est d'explorer le potentiel d'un SOLAP raster continu pour répondre aux besoins de la police, nous allons nous focaliser sur une méthode d'interpolation fréquemment utilisée dans le domaine du « crime mapping » : l'estimation de la densité par la méthode des noyaux (« Kernel Density Estimation » ou KDE). Le choix de cette méthode est motivé par trois raisons expliquées dans les prochains paragraphes.

Comme nous allons le voir à la section 7.3.3, KDE est une technique très populaire dans la littérature « crime mapping » pour sa capacité à déterminer les points chauds (hotspots) de criminalité de manière efficace et visuellement élégante (Eck *et al*, 2005; Chainey *et al*, 2008; Chainey, 2013; Hart & Zandbergen, 2014). Les hotspots calculés par

KDE permettent aux policiers d'analyser la distribution spatiale de la criminalité de manière cartographique¹⁴³, mais aussi de prévoir l'emplacement des futurs délits.

Ensuite, KDE est une méthode d'interpolation assez particulière dans le sens où elle produit des champs à partir de phénomènes discrets d'un point de vue spatial. Les données ainsi intégrées dans notre SOLAP présentent donc un caractère à la fois discret (les délits) et continu (les KDE construits à partir des délits) que nous pourrions exploiter dans la validation de notre modèle (chapitre 8). Avec un même jeu de données, nous serons en mesure de démontrer non seulement le potentiel d'un SOLAP raster sur des données continues, mais aussi de comparer ses performances avec un SOLAP vectoriel¹⁴⁴ sur des données discrètes. Notons qu'il existe d'autres méthodes populaires pour construire un champ continu à partir de données spatialement discrètes. Citons par exemple le modèle de potentiel fréquemment utilisé en géographie humaine (Donnay & Binard, 1991; Donnay & Unwin, 2001).

Finalement, la représentation de données discrètes sous formes de hotspots à différents niveaux d'échelle permet de minimiser un problème bien connu de la géographie : les unités spatiales modifiables (Openshaw, 1983; Taylor *et al*, 2003; Dark & Bram, 2007) ou MAUP (« Modifiable Areal Unit Problem »). Agréger des données sur des zones spatiales est inévitable pour les soumettre à une analyse géostatistique. Ces zones ont cependant un effet négatif sur l'analyse de l'utilisateur qui se retrouve biaisée par l'influence de frontières artificielles. La Figure 78 illustre ce problème sur différents types de découpages spatiaux¹⁴⁵. Le MAUP est d'autant plus présent lorsque les données sont agrégées sur des zones indépendantes des données elles-mêmes, telles que les entités administratives. En particulier pour les données criminelles, ne dit-on pas que « le crime n'a pas de frontière » ? La plupart des outils SOLAP vectoriels existants ne sont capables de représenter les mesures qu'en les agrégeant sur des entités prédéfinies et communes pour toutes les données : les membres spatiaux (que nous appelons « membres spatialisés »). Les résultats cartographiques sont donc très biaisés par le MAUP. D'un autre côté, lorsqu'une carte représente des hotspots dans notre SOLAP continu, les zones (hotspots) sont déterminées par les données elles-mêmes et la seule influence du MAUP provient des paramètres choisis dans l'interpolation KDE (Zaninetti, 2005). Comme nous le verrons, notre modèle offre la possibilité à l'utilisateur de faire varier ces paramètres avec son échelle d'analyse et donc de minimiser l'influence du MAUP.

¹⁴³ Le résultat d'une fonction KDE représenté de manière cartographique est appelé « carte de chaleur » (QGIS, 2015).

¹⁴⁴ Pour rappel, tous les SOLAP existants sur le marché sont vectoriels et aucun n'est capable de représenter des phénomènes spatialement continus.

¹⁴⁵ Remarquons que le MAUP intervient même sur un découpage spatial régulier (Figure 78c).

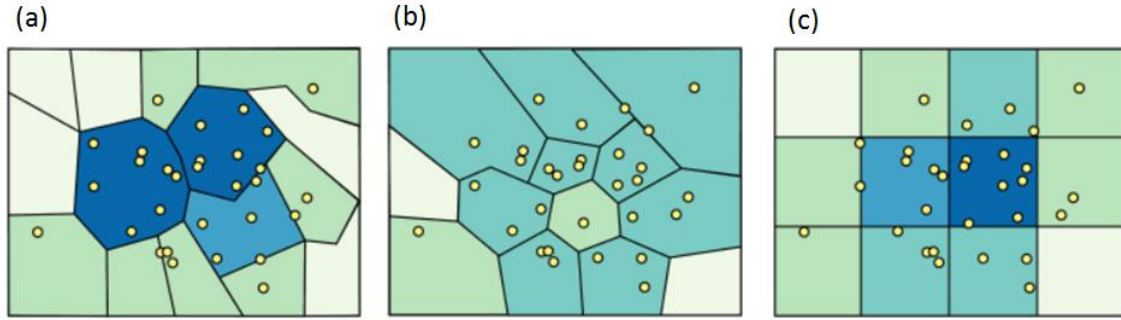


Figure 78. Illustration du problème des unités spatiales modifiables sur un découpage spatial irrégulier (a et b) et régulier (c) (Ervin, 2015)

7.3. Estimation de la densité par la méthode des noyaux (KDE)

L'objectif de cette section est de présenter la méthode KDE (De Smith *et al*, 2007) d'un point de vue théorique et d'analyser l'influence de ses paramètres dans les résultats cartographiques qu'elle produit. Nous aborderons ensuite son utilisation dans le domaine spécifique de la cartographie criminelle.

7.3.1. Champ KDE

Comme nous l'avons vu, KDE est une méthode d'interpolation générant un champ à partir de données discrètes. Ces données peuvent donc être initialement des entités vectorielles ponctuelles, linéaires ou zonales. Puisque notre objectif de validation du modèle SOLAP continu est de modéliser des délits représentés par des points, nous ne développerons pas le cas particulier des KDE appliqués sur des lignes ou des polygones. Nous admettons donc qu'une interpolation KDE transforme un nuage de points en champ. Le champ généré par un KDE correspond à une surface mathématique plus ou moins lissée que nous appelons « champ KDE ».

La valeur d'un champ KDE en un point j est inversement proportionnelle à la distance des points du nuage par rapport à j . Autrement dit, plus on a de points proches de j et plus la valeur du champ KDE sera élevée en j . Un exemple de fonction KDE couramment utilisée est la fonction quartique (Figure 79b). Celle-ci est donnée par la formule suivante :

$$p_j = \sum_{i=1}^n v_i * \frac{3}{r^2 * \pi} * \left[1 - \left(\frac{d_{ij}^2}{r^2} \right) \right]^2$$

Dans cette formule, p_j est la valeur du champ KDE en un point j et v_i est la variable associée à un point i dans un nuage comptant n points (imaginons par exemple une pondération associée à la gravité d'un délit). Si aucune variable n'est associée aux points du nuage, v_i a toujours la valeur 1 (imaginons par exemple que tous les délits soient considérés sur un même pied d'égalité). d_{ij} exprime la distance entre i et j . r est le rayon du cercle délimitant le voisinage de j dans lequel sont situés les n points ayant une

influence non-nulle sur la valeur du champ KDE en j ($d_{ij} \leq r$). Ce cercle, centré sur j , est appelé fenêtre de lissage (ou « bandwidth ») et son rayon est le même¹⁴⁶ pour tous les points du champ KDE. Une valeur de KDE en un point j est donc une somme de tous les points du nuage situés dans la fenêtre de lissage, pondérée de manière proportionnelle à l'inverse de leur distance au point j . La plupart des fonctions KDE dépendent uniquement de deux variables r et d , et elles peuvent donc être exprimées de manière générale par une fonction K de ce type :

$$p_j = \sum_{i=1}^n v_i * K(r, d_{ij})$$

Si aucune variable v n'est associée aux points du nuage, v a toujours la valeur 1 et par conséquent :

$$p_j = \sum_{i=1}^n K(r, d_{ij})$$

Les points voisins dans le résultat d'une fonction KDE sont spatialement autocorrélés puisqu'ils partagent logiquement les mêmes proches voisins dans le nuage de points. Le résultat d'un KDE appliqué sur les mailles d'une grille est donc bien un champ. Plus spécifiquement, un champ raster KDE est construit en alimentant un raster vide¹⁴⁷ par balayage de tous ses pixels en lignes et en colonnes. La valeur attribuée à un pixel est le résultat de la fonction K pour le point situé au centre du pixel¹⁴⁸ dans l'espace géographique. Supposons un raster avec N pixels, et un nuage A de a points. A chaque pixel p_j est attribuée la valeur $\sum_{i=1}^a K(r, d_{ij})$. On dénote par $K(A)$ le raster qui en résulte.

Il va de soi que l'emprise spatiale du raster doit être adaptée à celle du nuage de points, et qu'une trop faible résolution spatiale du raster par rapport à l'échelle d'analyse a un impact négatif sur l'analyse du champ KDE par l'utilisateur (cette réalité est valable pour n'importe quel raster de manière générale).

Notons que les valeurs d'un champ KDE ont tendance à être moins fiables lorsqu'on se rapproche des bords du nuage de points. A moins que le phénomène modélisé ne soit considéré comme absent à l'extérieur de la zone d'étude (imaginons par exemple que le territoire couvert par le nuage soit une île), les valeurs du champ KDE à proximité des bords sont en effet biaisées par l'absence de voisins à l'extérieur du nuage de points.

¹⁴⁶ Il existe certaines méthodes KDE particulières où la fenêtre de lissage varie avec j dans un même champ KDE.

¹⁴⁷ Avant le passage de l'algorithme KDE, toutes les cellules ont la valeur « no data ».

¹⁴⁸ Les coordonnées (y, x) du centre d'un pixel $p_{r,c}$ sont retrouvées en transformant les coordonnées $(r+0.5, c+0.5)$ de l'espace raster vers l'espace géographique. En effet, les coordonnées (r, c) attribuées à un pixel dans l'espace raster correspondent à son coin supérieur gauche.

7.3.2. Influence de la fenêtre de lissage et de la fonction KDE

La taille du « bandwidth » est un paramètre très important car il détermine le lissage global de la surface (ou champ) KDE. Plus la fenêtre de lissage est grande, plus les points éloignés ont une influence sur la valeur p_j , et plus le champ KDE est lissé. Autrement dit, une grande fenêtre montre les variations globales dans la distribution spatiale du nuage de points (pour une analyse à petite échelle), et une petite fenêtre en montre les variations locales (pour une analyse à grande échelle). Comme nous le verrons à la section 7.6.1.2, cette propriété de la fenêtre de lissage se traduit par une variation globale de la taille des hotspots déterminés sur base du champ KDE. Bien souvent, la taille de fenêtre optimale pour l'analyse de l'utilisateur est déterminée de manière empirique par une méthode « essai-erreur ».

La fonction quartique est une des plus populaires car, à ce jour, elle est la seule proposée dans le module « Spatial Analyst » du SIG-logiciel ArcGIS (ESRI, 2012). Des logiciels « open source » comme QGIS (2015), Grass GIS (2015) ou simplement gratuits comme CrimeStat (Levine, 2010) proposent pourtant un éventail de fonctions beaucoup plus large comme celles représentées dans la Figure 79. Parmi elles, la distribution uniforme (Figure 79a) est la seule fonction où le critère de distance est géré de manière complètement binaire : les points à l'intérieur de la fenêtre de lissage sont pondérés par 1 et les points en dehors sont pondérés par 0. Les autres fonctions sont caractérisées par des façons différentes de faire décroître l'impact des points voisins sur le résultat lorsqu'ils s'éloignent du centre de la maille (point j). Par exemple, l'éloignement des points du nuage par rapport à j a un impact plus important dans la fonction triangulaire (Figure 79c) que dans la fonction quartique (Figure 79b). Le choix d'une fonction KDE par rapport à une autre est donc tout aussi déterminant que le « bandwidth » puisque les différentes fonctions présentent elles-mêmes des lissages différents. Notons que la fonction normale (Figure 79d) est assez particulière car elle ne nécessite pas de « bandwidth ». Tous les points du nuage participent au calcul de toutes les valeurs des mailles. Cette fonction figure aussi parmi les plus populaires puisqu'elle s'adapte automatiquement à la distribution spatiale du nuage de points sans ambiguïté sur le choix de la taille de la fenêtre de lissage (lissage global). Néanmoins, son calcul est nettement plus long que les autres, justement à cause de l'absence de fenêtre. La fonction quartique est considérée comme une bonne approximation de la fonction normale (De Smith *et al*, 2007) lorsque la fenêtre est adaptée à la distribution spatiale globale (plusieurs méthodes de calcul de cette fenêtre sont proposées dans la littérature).

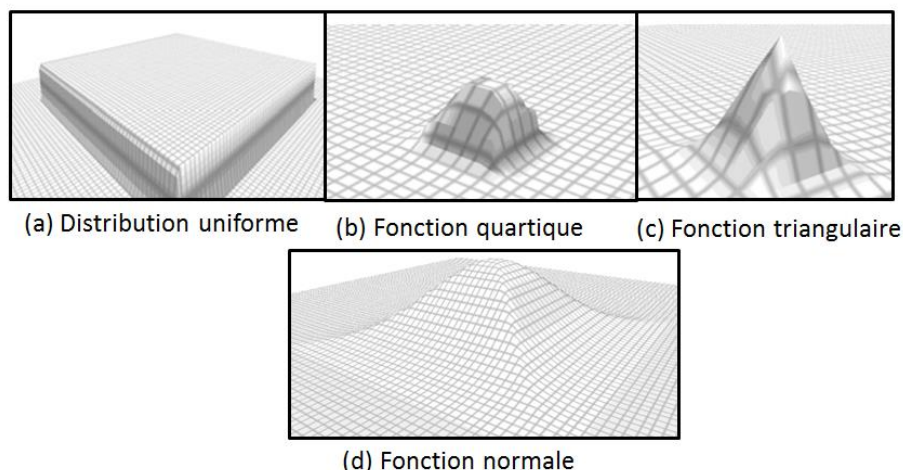


Figure 79. Exemples de fonctions KDE (Di Salvo et al, 2005)

Notons finalement que la distance euclidienne n'est pas la seule à être utilisée dans les interpolations KDE. La distance « réseau » est notamment très intéressante pour modéliser des phénomènes dépendant d'un réseau. Un exemple traitant des accidents de la route est présenté par Dabin *et al* (2013).

7.3.3. Détermination des hotspots de criminalité

Dans le domaine du « crime mapping », les cartes de hotspots sont fréquemment utilisées à un niveau stratégique pour analyser la distribution spatiale de la criminalité. En partant de l'hypothèse qu'un hotspot couvrant le ou les même(s) type(s) de délit(s) présente une certaine stabilité temporelle, ces cartes sont également utilisées dans un but préventif (Chainey *et al*, 2008; Chainey, 2013; Hart & Zandbergen, 2014). Les forces de Police sont alors concentrées dans les zones géographiques attribuées à des hotspots de criminalité pour la période la plus récente.

Plusieurs méthodes existent pour produire ce type de carte : ellipses de dispersion, cartes choroplèthes, cartes thématiques basées sur une grille raster (comme celles illustrées dans les chapitres 5 et 6), *etc.* L'interpolation KDE appliquée sur un raster est néanmoins considérée par de nombreux auteurs (McGuire & Williamson, 1999 ; Eck *et al*, 2005) comme la méthode produisant les meilleurs résultats. En effet, elle ne dépend pas de frontières artificielles (MAUP¹⁴⁹), tous les points contribuent à la détermination des hotspots et leur forme n'est pas trop influencée par la résolution spatiale de la grille (Chainey *et al*, 2008), pour autant que celle-ci ne soit pas trop faible par rapport à l'échelle d'analyse. Pour isoler les hotspots, la méthode la plus simple et la plus courante consiste à appliquer une classification par quantiles sur un champ KDE et de n'en garder que la ou les première(s) classe(s) (Figure 80). Par exemple, si un raster de 1000 pixels est soumis à une classification selon 5 quantiles fixés à 20%, un isolement de la première classe conservera les 200 pixels présentant les plus hautes valeurs KDE sur l'ensemble du raster.

¹⁴⁹ « Modifiable Areal Unit Problem ».

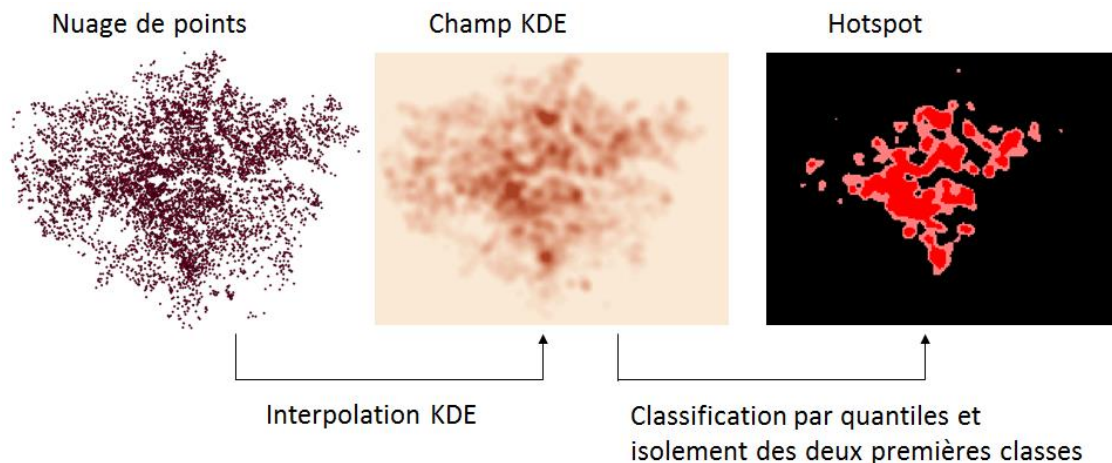


Figure 80. Construction de hotspots en estimant la densité par la méthode des noyaux (KDE)

Grâce à un indice évaluant le pouvoir prédictif des hotspots de criminalité (« Predictive Accuracy Index » ou PAI) introduit par Chainey *et al* (2008), plusieurs études ont démontré que le choix de la fonction KDE, mais surtout la taille de la fenêtre de lissage, a une influence significative sur la qualité des hotspots produits par KDE (Chainey, 2013; Hart & Zandbergen, 2014). Comme le montre la Figure 86, les hotspots ont tendance à être plus petits et plus nombreux lorsque la taille de la fenêtre diminue. Dans un contexte purement analytique de la distribution spatiale de la criminalité (SOLAP), celle-ci devrait donc être adaptée à l'échelle d'analyse.

Il n'existe hélas aucune méthode précise de détermination de la taille de la fenêtre de lissage qui aurait été approuvée par la communauté « crime mapping ». Pour une échelle d'analyse donnée, plusieurs calculs de KDE sont alors nécessaires pour déterminer la bonne taille de fenêtre de manière empirique. D'un autre côté, Chainey *et al* (2008) estiment qu'un rayon de fenêtre de lissage cinq fois plus grand que la taille d'un pixel dans l'espace géographique (ou inverse de la résolution spatiale) est suffisant pour que les hotspots ne soient pas trop « pixélisés ». Ainsi, si plusieurs couples de métadonnées « résolution spatiale/fenêtre de lissage » sont proposés dans les niveaux spatiaux pré-calculés (hiérarchie spatiale) d'un outil SOLAP intégrant des champs KDE, l'utilisateur devrait être en mesure de retrouver un paramétrage adapté à son échelle d'analyse par des opérations de forage spatial (donc en temps réel). Cet aspect sera illustré dans la section 7.6.1.2.

7.4. Intégration de champs KDE dans un hypercube raster

Nous venons de voir que KDE est une méthode d'interpolation très populaire, notamment en analyse criminelle (mais aussi dans d'autres domaines tels que l'écologie ou l'épidémiologie), car elles offrent une visualisation élégante et efficace de la distribution spatiale de grands jeux de données discrètes. De plus, leur utilisation minimise l'impact du MAUP (problème des unités spatiales modifiables) puisque les frontières des hotspots qu'elles produisent sont déterminées par les données elles-mêmes. L'idée d'exploiter cette technique dans un SOLAP raster permet non seulement de prouver son aptitude à gérer des champs continus, mais apporte surtout un support cartographique innovant dans le domaine

du SOLAP discret. En effet, la plupart¹⁵⁰ des outils SOLAP actuels ne sont capables d'agréger spatialement les données que sur des dimensions géographiques prédéfinies et communes à l'ensemble de l'hypercube (avec une influence considérable du MAUP). Comme nous allons le voir, l'exploitation du format raster dans notre modèle lui permet d'intégrer aisément ces champs KDE dans l'analyse de l'utilisateur. Les prochaines sections sont consacrées à la description et la démonstration des conditions de cette intégration.

7.4.1. Hypothèse SOLAP KDE

Pour intégrer des champs raster KDE, le modèle présenté aux deux chapitres précédents ne nécessite qu'un léger complément. Celui-ci repose sur une hypothèse que nous appelons « hypothèse SOLAP KDE » : soient deux nuages de points A et B et une fonction KDE K , l'agrégation non-spatiale par la somme de deux raster¹⁵¹ $K(A)$ et $K(B)$ issus du même niveau spatial d'un hypercube raster¹⁵² devrait satisfaire la condition suivante :

$$K(A) + K(B) = K(A \cup B)$$

Cette hypothèse signifie qu'en navigant¹⁵³ dans l'hypercube raster via l'interface cartographique (dimensions X , Y), l'utilisateur devrait avoir l'illusion¹⁵⁴ de calculer des champs KDE sur base d'un nuage de points défini par n'importe quel choix de combinaisons de membres non-spatiaux, alors que seules les mesures associées aux membres non-spatiaux détaillés ont été calculées par KDE au moment de l'intégration des données. Prenons par exemple un simple cube raster continu modélisant une dimension non-spatiale : les types de délits. Si l'utilisateur n'applique aucune coupe dans l'interface cartographique, l'agrégation non-spatiale des champs KDE représentant chaque type de délit (membre détaillé) résulterait en un champ KDE identique à celui construit sur le nuage de points caractérisé par tous les types de délits confondus. L'idée est la même pour n'importe quelle combinaison possible de coupes sur n'importe quelles dimensions non-spatiales modélisées dans un hypercube (cambriolages pour le premier trimestre, viols et homicides pour le premier semestre, etc.).

Comme nous allons le démontrer dans la section suivante, l'hypothèse SOLAP KDE est vérifiée à condition que les attributs « mesure ROLAP » d'un hypercube raster

¹⁵⁰ Dans certains outils SOLAP, il est également possible d'agréger des mesures spatiales, par exemple par l'union. Les frontières des entités représentées par la carte ne sont donc plus prédéfinies dans une dimension, mais le résultat n'est pas aussi efficace que dans un KDE qui tient compte de la distance de tous les points situés dans le voisinage de l'entité pour représenter un hotspot spatialement continu.

¹⁵¹ Rappel de la section 7.3.1: supposons un raster avec N pixels, et un nuage A de a points. A chaque pixel p_j est attribuée la valeur $\sum_{i=1}^a K(r, d_{ij})$. On dénote par $K(A)$ le raster qui en résulte.

¹⁵² Autrement dit : une tranche d'une mesure ROLAP atomique (cube raster) perpendiculaire à l'axe de la dimension non-spatiale B . Pour rappel, un attribut « mesure ROLAP » définit soit l'hypercube de base, soit un hypercuboïde matérialisé exprimant un niveau hiérarchique spatial donné.

¹⁵³ Pour autant que la fonction d'agrégation utilisée soit la somme ou la moyenne et que l'utilisateur n'effectue pas une opération de comparaison (section 4.2.7).

¹⁵⁴ A la différence près que l'opération dans le SOLAP raster est nettement plus rapide si l'interpolation KDE équivalente implique un grand nombre de points.

continu (voir section 6.4) soient contraints par deux métadonnées supplémentaires : la fonction KDE et la taille de la fenêtre de lissage¹⁵⁵.

7.4.2. Démonstration de l'hypothèse SOLAP KDE

Si deux nuages de points A et B sont deux ensembles non-ordonnés et exprimés par

$$A = \{e_i\}_{i=1}^a$$

$$B = \{e_i\}_{i=a+1}^b$$

On peut dire que :

$$A \cup B = \{e_i\}_{i=1}^{a+b}$$

Puisque deux raster issus d'un même niveau spatial d'un hypercube raster partagent les mêmes métadonnées spatiales et dimensionnelles, chaque pixel dans le raster $K(A)$ présente un pixel homologue dans le raster $K(B)$. Ces pixels homologues partagent à la fois les mêmes coordonnées spatiales dans l'espace raster et dans l'espace géographique, et on peut donc dire que deux pixels homologues sont tous les deux centrés sur un même point j . Puisqu'une agrégation non-spatiale par la somme de deux raster $K(A)$ et $K(B)$ consiste¹⁵⁶ à sommer les valeurs de chaque couple de pixels homologues, la valeur d'un pixel résultant de cette agrégation peut être formulée par :

$$p_j' = \sum_{i=1}^a K(d_{ij}, r) + \sum_{i=a+1}^b K(d_{ij}, r)$$

D'un autre côté, la valeur d'un pixel d'un raster $K(A \cup B)$ peut être exprimée par :

$$p_j = \sum_{i=1}^{a+b} K(d_{ij}, r)$$

Puisque la somme est associative et distributive, dans le cas où r (le rayon de la fenêtre de lissage) et K (la fonction KDE) sont identiques dans les trois raster, on peut donc dire que :

$$p_j' = p_j$$

Puisque deux raster issus d'un même niveau spatial d'un hypercube partagent les mêmes métadonnées spatiales et dimensionnelles, ils sont composés exactement du même nombre de pixels et les pixels de $K(A)$ ont tous un homologue dans $K(B)$. Par conséquent :

$$K(A) + K(B) = K(A \cup B)$$

¹⁵⁵ Cela exclut donc la fonction normale (en raison de l'absence de fenêtre de lissage) et tous les algorithmes KDE à fenêtre adaptative (variant d'un point à l'autre de l'espace).

¹⁵⁶ Pour rappel, une agrégation non-spatiale est une opération locale d'algèbre de carte.

7.4.3. Conséquence de l'hypothèse SOLAP KDE

En suivant un raisonnement similaire à celui de la section précédente, on s'aperçoit que lorsque l'hypothèse SOLAP KDE est vraie, elle fonctionne pour les fonctions KDE dépendant uniquement du critère distance, mais aussi pour les fonctions KDE tenant compte d'une variable v associée aux points dont on estime la densité.

Supposons un raster avec N pixels, et un nuage A de a points. A chaque pixel p_j est attribuée la valeur $\sum_{i=1}^a v_i * K(d_{ij}, r)$. On dénote $K(v, A)$ le raster qui en résulte. Si la variable v présente une valeur constante v_A pour un ensemble de points A et une valeur constante v_B pour un ensemble de points B , on peut dire que¹⁵⁷ :

$$v_A * K(A) + v_B * K(B) = K(v, A \cup B)$$

Cela signifie que lorsqu'un hypercube raster est soumis à une agrégation non-spatiale selon une somme pondérée par une variable caractérisant les membres d'une dimension non-spatiale, le résultat est un champ raster KDE dont la pondération tiendrait compte de cette variable en plus du critère distance. Par exemple, si on associe la valeur 1 aux vols sans violence et la valeur 2 aux vols avec violence, le résultat de la somme pondérée par ces valeurs en fonction de la dimension « type de délit » dans l'hypercube raster est le même que celui d'une fonction KDE accordant deux fois plus d'influence aux vols avec violence qu'à ceux sans violence. Cette conséquence de l'hypothèse SOLAP KDE est démontrée dans la section suivante.

7.4.4. Démonstration de la conséquence de l'hypothèse SOLAP KDE

Si la première hypothèse SOLAP KDE est vraie :

$$K(A) + K(B) = K(A \cup B)$$

L'égalité suivante est aussi vraie :

$$K(v, A) + K(v, B) = K(v, A \cup B)$$

En effet, cette égalité se vérifie sur chaque pixel homologue de ces trois raster puisque r est constant et K est la même fonction KDE :

$$\sum_{i=1}^a v_i * K(d_{ij}, r) + \sum_{i=a+1}^b v_i * K(d_{ij}, r) = \sum_{i=1}^{a+b} v_i * K(d_{ij}, r)$$

¹⁵⁷ Multiplier un raster par une constante signifie multiplier la valeur de tous ses pixels par cette constante.

Puisque v a une valeur constante v_A sur tout l'ensemble A , la valeur d'un pixel de $K(v, A)$ est donnée par :

$$\sum_{i=1}^a v_i * K(d_{ij}, r) = v_A * \sum_{i=1}^a K(d_{ij}, r)$$

Et ceci est vrai pour tous les pixels de $K(v, A)$:

$$K(v, A) = v_A * K(A)$$

Par un même raisonnement, on peut donc dire que :

$$K(v, B) = v_B * K(B)$$

Par conséquent :

$$v_A * K(A) + v_B * K(B) = K(v, A \cup B)$$

7.4.5. Métadonnées des mesures ROLAP KDE

Lorsqu'un hypercube raster continu intègre des champs KDE, une valeur de champ est associée à la maille d'une mesure raster de l'hypercube. Puisqu'une mesure ROLAP définit un ensemble de mesures raster (mailles) selon les dimensions spatiales X , Y (ou C , R) et éventuellement une dimension non-spatiale B (cube raster), une mesure ROLAP définit donc un champ raster KDE (absence de dimension B) ou une couverture spatiale¹⁵⁸ KDE caractérisée par un vecteur de N_B valeurs d'attribut (nombre de membres de la dimension B).

Comme nous venons de le démontrer, pour que l'hypothèse SOLAP KDE soit vraie, un attribut « mesure ROLAP KDE » doit être contraint par deux métadonnées supplémentaires par rapport à un hypercube raster quelconque : la fonction KDE (K) et la taille de la fenêtre de lissage (r). Comme le montre la Figure 81, ces métadonnées supplémentaires sont définies dans la table des métadonnées des mesures ROLAP (qui avait été introduite dans la section 6.4).

Rappelons qu'une mesure ROLAP est strictement associée à un niveau de hiérarchie spatiale (hiérarchie matérialisée). Il y a donc autant d'attributs « mesures ROLAP KDE » dans la table des faits que de niveaux spatiaux différents¹⁵⁹.

¹⁵⁸ Voir section 2.2.

¹⁵⁹ Eventuellement, si la hiérarchie de la dimension B est matérialisée, il y a autant d'attributs « mesure ROLAP » différents que d'éléments dans le produit cartésien des niveaux spatiaux par les niveaux de B .

Table des métadonnées	
Identification de la mesure	Table des faits
	Mesure ROLAP KDE
Métadonnées spatiales	SCR
	Paramètres de translation (t_R, t_C)
	Paramètres d'homothétie (h_R, h_C)
Métadonnées dimensionnelles	N_R
	N_C
	N_B
Métadonnées KDE	Fonction KDE (K)
	Fenêtre de lissage (r)

Figure 81. Table des métadonnées d'un hypercube raster KDE

7.5. Exemple d'hypercube raster KDE

Dans cette section, un exemple d'hypercube raster KDE est présenté afin d'illustrer les opérations SOLAP décrites lors des prochaines sections. Cet hypercube (Figure 82) est assez proche de l'exemple de la section 6.5 : il est basé sur les mêmes données de criminalité londonienne (UK Police, 2014) selon les mêmes instances de dimensions non-spatiales : une dimension B temporelle couvrant l'année 2012 hiérarchisée en mois et en trimestre, et une dimension ROLAP « type de délit » non-hiérarchisée. Par contre, la hiérarchie spatiale (dimensions X, Y) est légèrement différente car le niveau « pixel = 1000 m » donne d'assez mauvais résultats lorsqu'il est représenté par une mesure KDE (la fenêtre de lissage adaptée à cette résolution engendre des hotspots beaucoup trop grands, même pour une petite échelle d'analyse). Pour cet exemple, les niveaux spatiaux sont donc « pixel = 100 m » et « pixel = 300 m ». Concernant les fenêtres de lissage, nous les avons adaptées à la résolution spatiale par la méthode conseillée par Chainey *et al* (2008) : une fenêtre de lissage cinq fois plus grande que la taille des pixels dans l'espace géographique. Cela donne donc un niveau « pixel = 100 m ; fenêtre de lissage = 500m » et un niveau « pixel = 300 m ; fenêtre de lissage = 1500 m ». La fonction KDE utilisée pour les deux niveaux est la même¹⁶⁰ que celle proposée par le SIG-logiciel ArcGIS (ESRI, 2012) : la fonction quartique.

Conceptuellement, l'hypercube présenté ici comprend deux mesures : le nombre de délits et la densité de délits. Le nombre de délits est la même mesure que celle de l'hypercube de la section 6.5 (nombre de délits à l'intérieur de l'emprise spatiale d'un pixel). En revanche, la densité de délits est une nouvelle mesure exprimant la valeur du champ raster KDE. La continuité spatiale ne peut donc être représentée que via cette mesure dans l'interface cartographique. D'un point de vue logique, puisque deux mesures

¹⁶⁰ Remarquons que l'algorithme KDE utilisé n'est pas celui d'ArcGIS, mais un que nous avons implémenté de toutes pièces dans le SGBD PostGIS (fonction PL/PGSQL). Visuellement, les résultats sont très similaires à ceux produits par l'algorithme d'ESRI.

distinctes sont associées à une hiérarchie spatiale matérialisée¹⁶¹, la table des faits résultant de ce schéma comprend quatre mesures (deux mesures pour deux niveaux spatiaux).

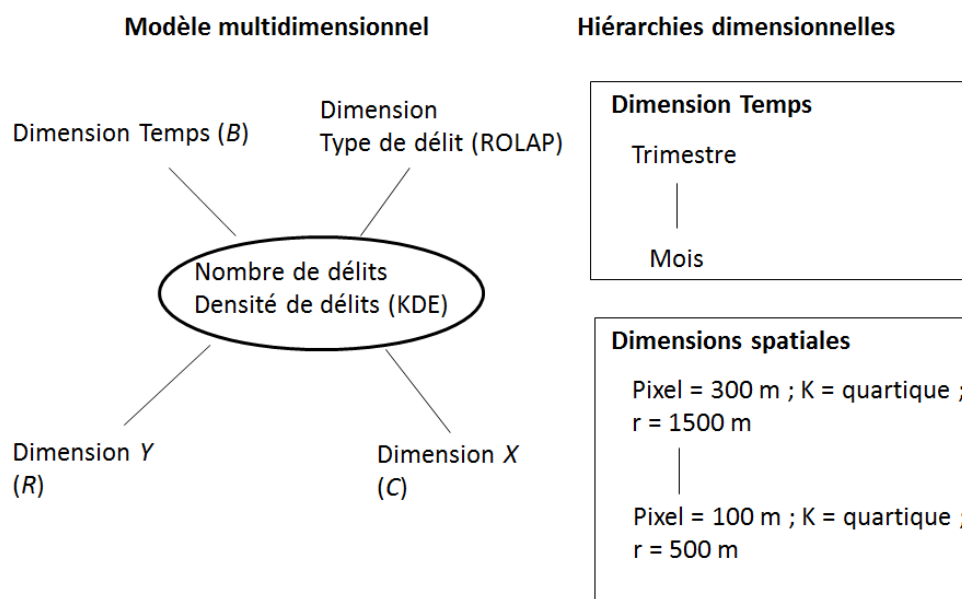


Figure 82. Exemple de schéma conceptuel d'un hypercube raster KDE

Notons que pour un utilisateur lambda, les valeurs d'un champ KDE dans un SOLAP sont intéressantes d'un point de vue purement relatif. Il est en effet aisé d'interpréter la variabilité de ce genre de mesure en fonction d'une ou plusieurs dimensions (en particulier les dimensions spatiales sur une carte), mais une valeur absolue (qui techniquement, exprime une fréquence de distances) est nettement moins parlante pour la plupart des utilisateurs non sensibilisés à l'analyse spatiale. Ces derniers devraient la percevoir comme un indice augmentant avec le nombre de délits dans un espace géographique plus ou moins étendu, en fonction de l'échelle d'analyse. Dans le contexte du « crime mapping », l'utilisateur devrait donc toujours avoir la possibilité d'afficher aussi bien les mesures KDE que les nombres de délits, peu importe l'échelle d'analyse, en particulier dans une interface graphique ou tabulaire.

7.6. Opérations SOLAP sur champs raster KDE

Dans les prochaines sections, nous allons décrire quelques opérations SOLAP orientées utilisateur lorsqu'elles impliquent des champs raster KDE. Celles-ci sont appliquées sur un hypercube raster KDE implémenté selon le schéma conceptuel présenté à la section 7.5. Comme pour tout hypercube raster, ces fonctions SOLAP exploitent la chaîne d'opérations élémentaires décrite aux sections 5.7 et 6.6 (elles sont toujours les mêmes peu importe les types de données raster intégrées, discrètes ou continues). La section 7.6.1 est consacrée aux opérations appliquées sur l'interface cartographique et la section 7.6.2, beaucoup plus courte car moins importante dans ce contexte, est consacrée aux opérations appliquées sur l'interface graphique.

¹⁶¹ Pour rappel : la hiérarchie spatiale est toujours matérialisée.

7.6.1. Interface cartographique

Afin de rester cohérent vis-à-vis des précédentes illustrations des opérations SOLAP raster, seule la fonction d'agrégation « somme » est employée dans les prochains exemples. Il est cependant important de préciser que notre modèle fonctionne avec à peu près n'importe quelle fonction d'agrégation¹⁶² et qu'il est généralement plus pertinent d'agréger un champ continu par la moyenne (par exemple, la moyenne mensuelle des températures plutôt que la somme). Puisque les valeurs absolues de champs KDE nous intéressent moins que les nombres de délits, agréger (spatialement ou non) une mesure KDE par la moyenne ou par la somme revient exactement au même d'un point de vue relatif : la variation relative de la mesure sur une carte ou sur un graphique est donc toujours la même, qu'elle résulte d'une somme ou d'une moyenne.

7.6.1.1. Opérations de coupe et de comparaison

Comme nous l'avons vu à la section 7.4.1, lorsque l'hypothèse SOLAP KDE est vraie (par la présence de contraintes spécifiques aux KDE sur les mesures ROLAP), toutes les opérations possibles de coupes non-spatiales appliquées sur des champs raster KDE entraînent un résultat similaire à celui-ci d'un KDE construit sur le nuage de points défini par ces coupes. La Figure 83 illustre ainsi une coupe des cambriolages sur le premier trimestre 2012 (Figure 83a, Figure 83b) et une coupe des cambriolages sur le second trimestre 2012 (Figure 83c, Figure 83d). Sur les Figure 83a et Figure 83c, la légende est un étirement linéaire pour visualiser les champs KDE, et sur les Figure 83b et Figure 83d, la légende est une classification du champ KDE par quantiles de 12,5 %. Sur les huit classes générées par cette dernière méthode ($8 * 12,5 \% = 100 \%$), seule la première classe est conservée afin d'isoler les hotspots : les zones rouges représentent donc les 12,5 % de l'ensemble de la zone d'étude (Londres et sa périphérie) présentant la plus forte densité de cambriolages. Puisque ces cartes ont été calculées à partir du niveau spatial « Pixel = 300 m ; $r = 1500$ m ; $K = \text{quartique}$ », le résultat est l'équivalent d'un KDE appliqué sur les nuages de points définis par les coupes non-spatiales avec les mêmes paramètres que ceux du niveau spatial.

Pour obtenir le résultat de la Figure 83, la chaîne d'opérations élémentaires a été appliquée de la manière suivante (dans l'ordre de la Figure 75) :

¹⁶² Pour autant qu'elle soit distributive ou algébrique (comptage, somme, moyenne, *etc.*). Les fonctions holistiques (en particulier l'étendue et l'écart-type) fonctionnent sous certaines conditions dont nous discuterons dans le prochain chapitre.

- Filtrage vertical ROLAP : aucun
- Filtrage vertical raster : mesure ROLAP « densité de délits : pixel = 300 m ; $r = 1500$ m ; $K = \text{quartique}$ »
- Filtrage horizontal ROLAP : membre « cambriolage »
- **Filtrage horizontal raster :**
 - o Non-spatial : membre « trimestre 1 » (Figure 83a et Figure 83b)
 - o Non-spatial : membre « trimestre 2 » (Figure 83c et Figure 83d)
- Agrégation partielle ROLAP : aucune
- Agrégation partielle raster : aucune
- Agrégation totale ROLAP : aucune
- Agrégation totale raster :
 - o Non-spatiale (carte)

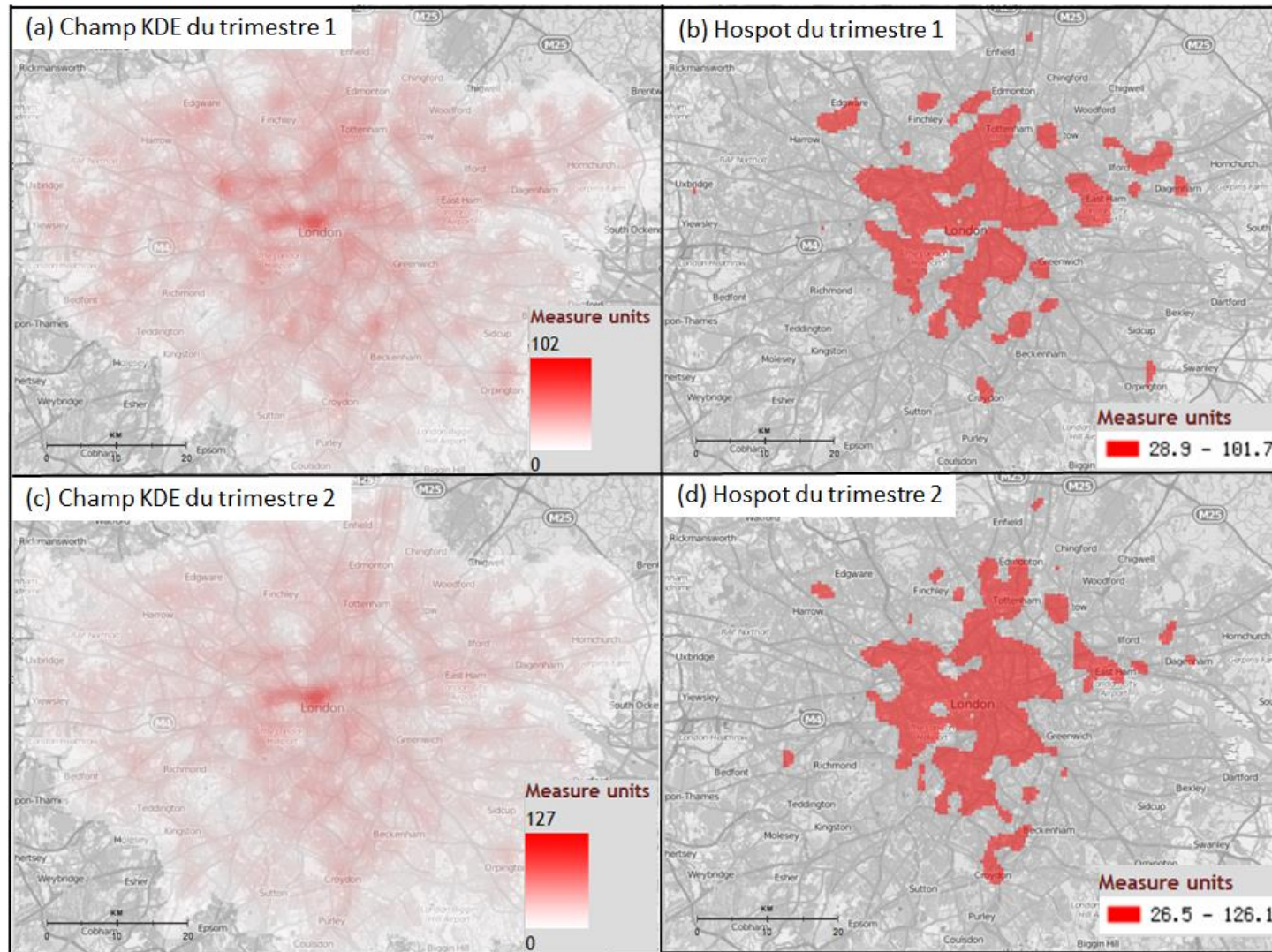


Figure 83. Opérations de coupe dans un hypercube raster KDE

La Figure 84 illustre le résultat d'une comparaison cartographique entre la coupe de la Figure 83a et la coupe de la Figure 83c. Il s'agit d'une carte montrant l'évolution de la densité de cambriolages entre le premier trimestre 2012 et le second trimestre 2012. Rappelons que dans la comparaison cartographique, le résultat d'une première coupe (premier trimestre) est soustrait au résultat d'une seconde coupe (deuxième trimestre) par opération locale d'algèbre de carte.

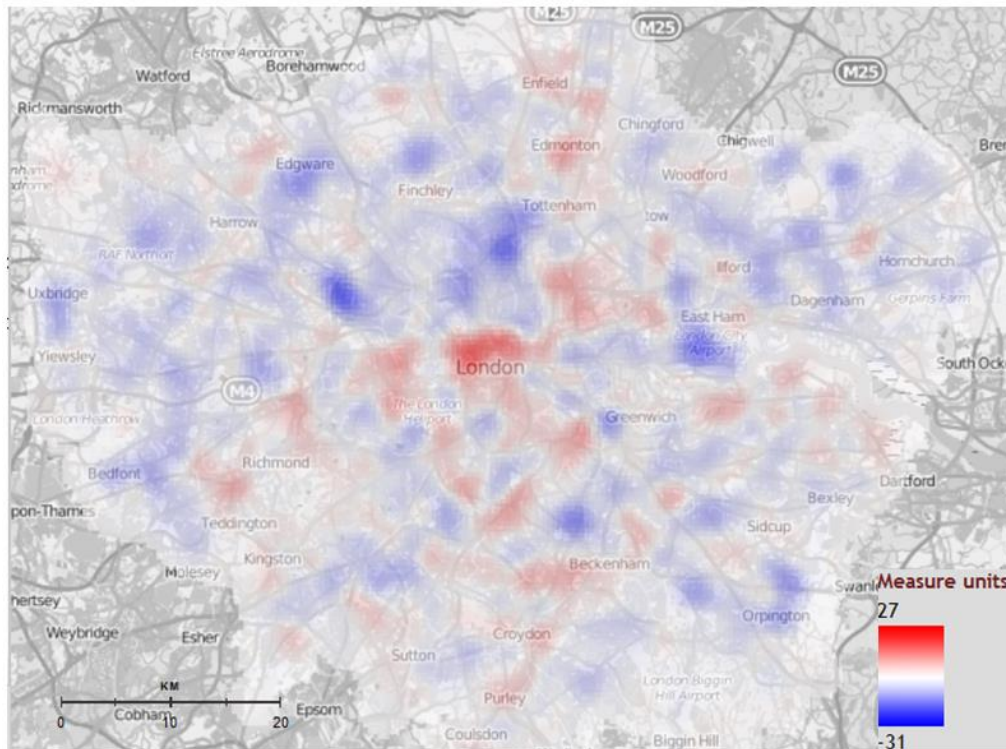


Figure 84. Opération de comparaison cartographique dans un hypercube raster KDE

La Figure 85 illustre le résultat d'une coupe spatiale sur un champ raster KDE. Dans la Figure 85a, une simple coupe non-spatiale sur les vols à l'étagère du premier trimestre est effectuée pour l'ensemble du territoire. Puisque la plupart de ces délits sont concentrés dans le centre-ville, l'opération SOLAP est complétée dans la Figure 85b par une coupe spatiale à partir d'un membre spatialisé représentant uniquement la périphérie (un « trou » est donc digitalisé dans le polygone modélisant le membre spatialisé « périphérie » pour exclure le centre de Londres). Le résultat de la coupe spatiale révèle ainsi beaucoup plus clairement les hotspots en dehors du centre puisque celui-ci n'influence plus l'étirement linéaire de la légende. Il s'agit d'une alternative¹⁶³ plus simple et plus directe au calcul d'une classification adaptée à un champ qui présente des régions à valeurs très élevées par rapport au reste du territoire (tel que le centre-ville). Notons que les coupes spatiales ne souffrent pas du phénomène de bords des champs KDE puisque les mesures aux alentours des frontières de la coupe sont toujours bien influencées par celles situées en dehors de la coupe (proximité du centre-ville dans la Figure 85b). En effet, elles résultent d'un découpage de champs qui ont été calculés à partir des points de l'ensemble du territoire couvert par l'hypercube. Évidemment, les mesures à proximité des bords de ce

¹⁶³ Avec un autre avantage important : les coupes réduisent les temps de traitement.

territoire, quant à elles, sont effectivement biaisées puisqu'aucun calcul n'a tenu compte des délits situés à l'extérieur de la périphérie londonienne au moment de l'intégration des données.

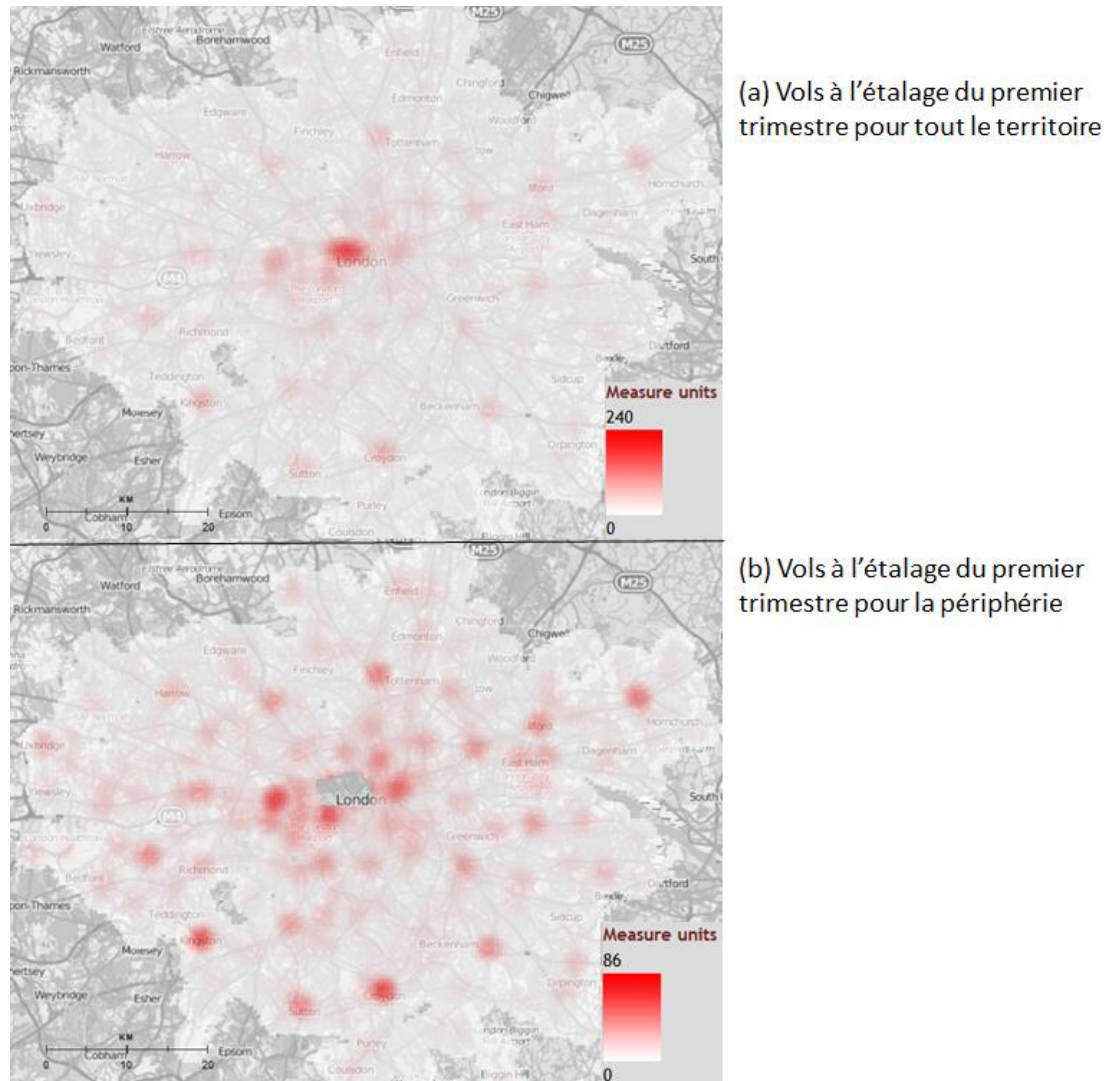


Figure 85. Opération de coupe spatiale dans un hypercube raster KDE

Pour obtenir le résultat de la Figure 85, la chaîne d'opérations élémentaires a été appliquée de la manière suivante (dans l'ordre de la Figure 75) :

- Filtrage vertical ROLAP : aucun
- Filtrage vertical raster : mesure ROLAP « densité de délits : pixel = 300 m ; $r = 1500$ m ; $K = \text{quartique}$ »
- Filtrage horizontal ROLAP : membre « vols à l'étalage »
- **Filtrage horizontal raster :**
 - o Non-spatial : membre « trimestre 1 »
 - o Spatial : raster dimensionnel comprenant un membre « Périphérie de Londres » (Figure 85b uniquement)
- Agrégation partielle ROLAP : aucune
- Agrégation partielle raster : aucune
- Agrégation totale ROLAP : aucune
- Agrégation totale raster :
 - o Non-spatiale (carte)

7.6.1.2. Opérations de forage spatial

Les opérations de forage spatial appliquées sur une mesure exprimant la densité par KDE sont assez particulières puisqu'elles permettent à l'utilisateur de faire varier les paramètres spécifiques aux champs KDE au moment de l'analyse SOLAP. Si plusieurs niveaux spatiaux ont été calculés au moment de l'intégration des données dans l'entrepôt, l'utilisateur bénéficie ainsi d'une certaine liberté pour adapter ces paramètres à son analyse de manière empirique. Puisque tous les niveaux spatiaux sont pré-calculés, les forages spatiaux sont très rapides et, comme nous l'avons déjà vu, les hypercuboides matérialisant les niveaux spatiaux les plus grossiers ont tout intérêt à être privilégiés dans les opérations appliquées sur de larges territoires, qu'elles aient lieu sur l'interface cartographique ou graphique (en particulier dans les entrepôts de données comptant un grand nombre de membres/dimensions). A l'inverse, les opérations appliquées sur les niveaux plus fins devraient être combinées à une coupe spatiale afin de réduire le territoire d'analyse et d'optimiser les temps de traitement. Dans l'exemple de la Figure 86, cette règle n'a pas été appliquée pour mieux illustrer l'opération basique de forage spatial sur une mesure KDE. Le forage dont il est question est appliqué sur une coupe non-spatiale impliquant le type de délit « troubles de l'ordre public » et le premier trimestre 2012. Dans la Figure 86a, les hotspots résultant d'une classification par quantiles (première des huit classes) ont été calculés à partir de la mesure « pixel = 300 m ; $r = 1500$ m ». Dans la Figure 86b, les hotspots isolés par la même classification ont été calculés à partir de la mesure « pixel = 100 m ; $r = 500$ m ». La Figure 86a est donc un « spatial roll up » et la Figure 86b un « spatial drill down ». Comme expliqué à la section 7.3.2, on y voit très clairement que la diminution de la taille de la fenêtre de lissage (« drill down ») engendre des hotspots plus petits et plus nombreux, et donc adaptés à une échelle d'analyse plus grande. En réalité, le « spatial roll up » s'apparente très fort à une opération de généralisation cartographique.

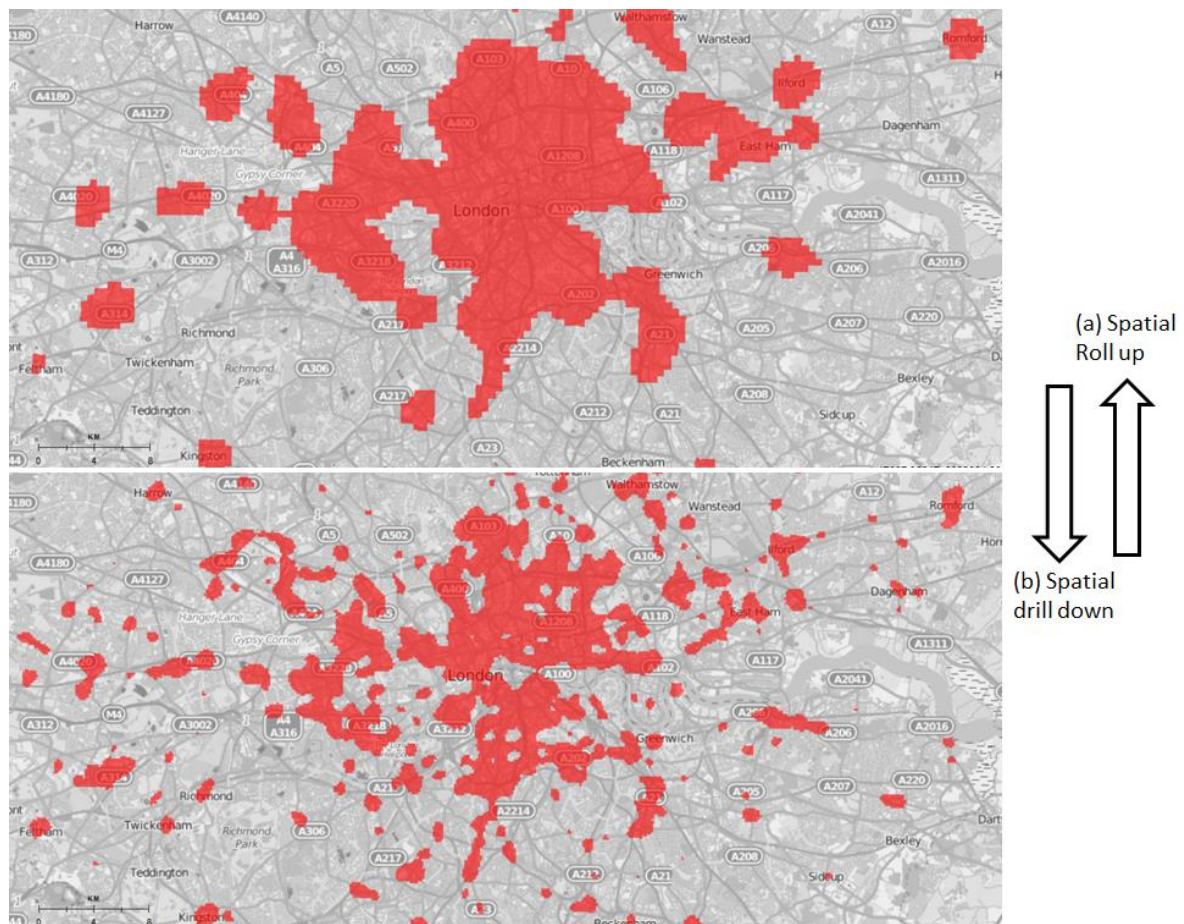


Figure 86. Opération de forage spatial dans un hypercube raster KDE

Pour obtenir le résultat de la Figure 86, la chaîne d'opérations élémentaires a été appliquée de la manière suivante (dans l'ordre de la Figure 75) :

- Filtrage vertical ROLAP : aucun
- **Filtrage vertical raster :**
 - o mesure ROLAP « densité de délits : pixel = 300 m ; $r = 1500$ m ; $K =$ quartique » (Figure 86a)
 - o mesure ROLAP « densité de délits : pixel = 100 m ; $r = 500$ m ; $K =$ quartique » (Figure 86b)
- Filtrage horizontal ROLAP : membre « troubles de l'ordre public »
- Filtrage horizontal raster :
 - o Non-spatial : membre « trimestre 1 »
- Agrégation partielle ROLAP : aucune
- Agrégation partielle raster : aucune
- Agrégation totale ROLAP : aucune
- Agrégation totale raster :
 - o Non-spatiale (carte)

7.6.2. Interface graphique

Le principal intérêt d'intégrer des champs KDE dans un entrepôt de données raster réside dans leur représentation cartographique. La visualisation de dimensions non-spatiales à travers des graphiques est en effet plus pertinente avec la mesure « nombre de délits » puisque les valeurs absolues sont plus compréhensibles par l'utilisateur. Néanmoins, puisqu'un SOLAP raster est capable d'intégrer tous les types de champs continus, il nous semble important d'illustrer les résultats que l'interface graphique est capable de produire avec ce genre de données. Une comparaison a donc été effectuée entre une agrégation par la moyenne de la mesure continue « densité de délits » résultant d'un KDE (Figure 87a) et une agrégation par la somme de la mesure discrète « nombre de délits » (Figure 87b). Les deux dimensions non-spatiales « type de délit » et « mois » sont affichées par les graphiques à partir d'une coupe sur les membres « vols à l'étalage », « cambriolages » et « troubles de l'ordre public » de la dimension « type de délit ». Aucune coupe spatiale n'est effectuée et les valeurs sont donc agrégées sur l'ensemble du territoire d'étude (Londres et sa périphérie). On peut logiquement observer que les valeurs relatives fournies par les champs KDE (Figure 87a) sont exactement les mêmes que celles fournies par le nombre de délits par pixel (Figure 87b) selon les deux dimensions affichées. Lorsqu'une coupe spatiale est effectuée, les valeurs relatives des deux mesures sont également très similaires, mis à part que les champs KDE se retrouvent légèrement influencés par les délits situés à l'extérieur de la coupe¹⁶⁴.

Pour obtenir le résultat de la Figure 87, la chaîne d'opérations élémentaires a été appliquée de la manière suivante (dans l'ordre de la Figure 75) :

- Filtrage vertical ROLAP : aucun
- **Filtrage vertical raster :**
 - o mesure ROLAP « densité de délits : pixel = 300 m ; $r = 1500$ m ; $K =$ quartique » (Figure 83a)
 - o mesure ROLAP « nombre de délits : pixel = 300 m » (Figure 83b)
- Filtrage horizontal ROLAP : membres « vols à l'étalage », « cambriolages » et « troubles de l'ordre public »
- Filtrage horizontal raster : aucun
- Agrégation partielle ROLAP : aucune
- Agrégation partielle raster : aucune
- Agrégation totale ROLAP : aucune
- Agrégation totale raster :
 - o spatiale (graphique)

¹⁶⁴ Pour autant que la taille du filtre spatial soit significativement plus importante que celle de la fenêtre de lissage de la mesure KDE. Par exemple, effectuer une coupe spatiale au moyen d'un cercle de 1 km de rayon n'a pas beaucoup de sens si la fenêtre de lissage fait également 1 km (une fenêtre de lissage de maximum 100 m serait plus adaptée pour ne pas trop tenir compte des valeurs extérieures à la coupe).

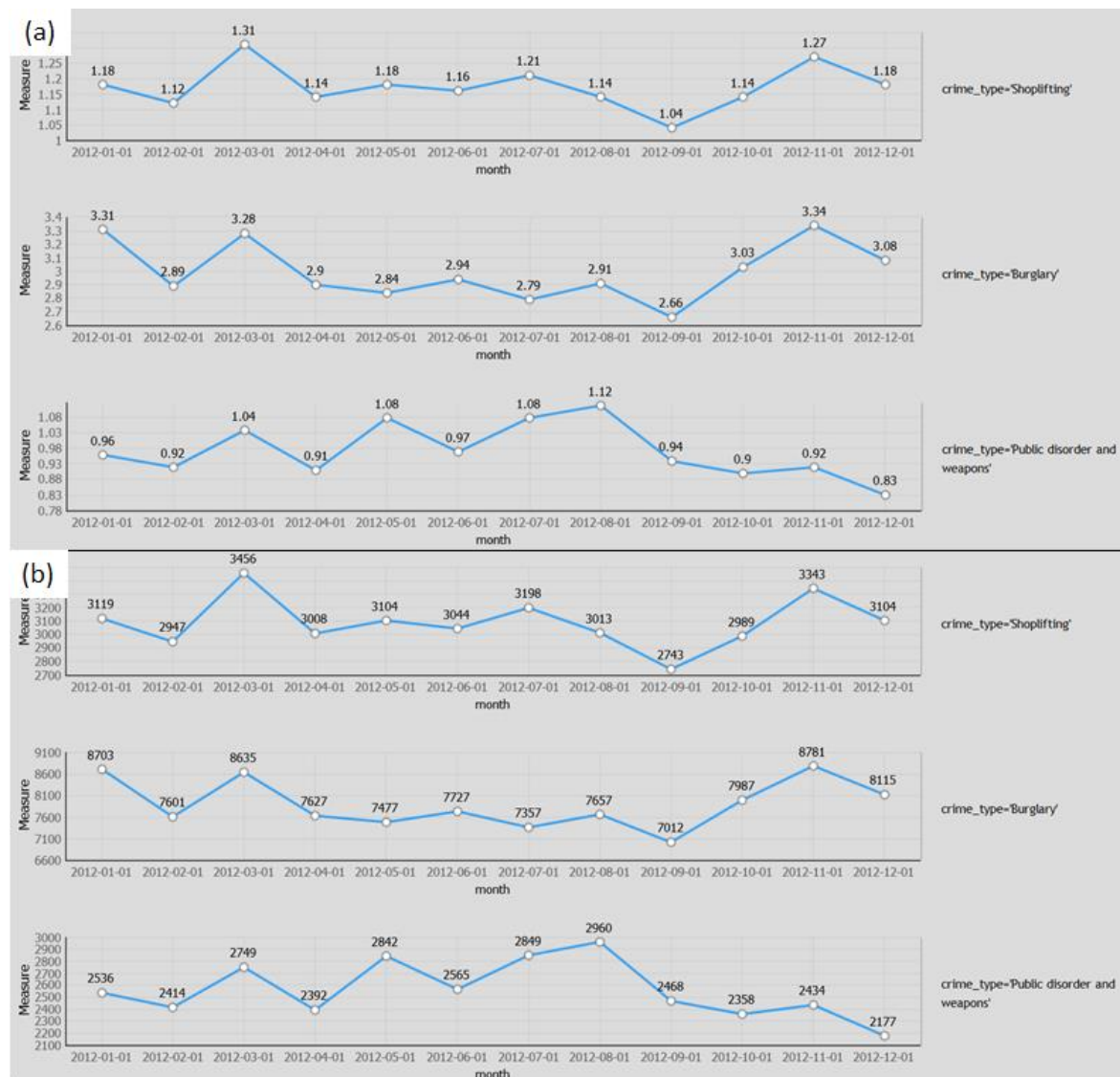


Figure 87. Comparaison entre les mesures « densité de délits » (a) et « nombre de délits » (b) dans l'interface graphique

7.7. Conclusion

Ce chapitre avait pour but d'étudier l'intégration de l'information spatialement continue dans le modèle d'hypercube raster développé dans le cadre de cette thèse. Comme nous l'avons vu dans le chapitre 2, la structure maillée des données raster permet une modélisation simple et efficace des phénomènes spatialement continus (on parle alors de champs raster). Puisque toutes les cellules d'un hypercube raster correspondent en réalité aux pixels d'un raster, le modèle développé lors des précédents chapitres était déjà bien adapté à ce type de phénomène. Un hypercube raster continu modélise donc une couverture spatiale continue où la mesure d'un fait détaillé est une valeur de champ définie par les membres de dimensions (spatiales et non-spatiales) associés au fait. Les calculs des faits non-détaillés sont effectués uniquement par agrégation des champs raster intégrés à l'entrepôt. Contrairement aux modèles SOLAP continus exploitant le format vectoriel, tous les calculs d'interpolation des champs sont réservés uniquement à la phase d'intégration des données et non à la phase d'analyse.

La suite du chapitre était focalisée sur l'intégration de champs raster résultant d'une méthode d'interpolation bien particulière : l'estimation de la densité par la méthode des noyaux (champs KDE¹⁶⁵). Ce choix était d'abord motivé par la popularité de la méthode dans le domaine de la cartographie criminelle pour produire des cartes de hotspots. Ces cartes sont effectivement très utilisées par la police à un niveau stratégique (ou décisionnel), notamment pour prévoir l'emplacement des futurs délits. D'un point de vue plus théorique, KDE est une méthode d'interpolation spatiale originale car elle produit une vision spatialement continue de phénomènes discrets (tels que les délits). Le choix de cette méthode s'avère donc pertinent pour valider l'intégration de champs dans notre modèle, mais surtout pour apporter un type d'interface cartographique original dans le domaine du SOLAP appliqué sur des phénomènes spatialement discrets. Ce nouveau support visuel des données spatiales a ainsi l'avantage de minimiser l'impact du problème des unités spatiales modifiables (MAUP¹⁶⁶) puisque les frontières d'un hotspot résultant d'un KDE sont définies à partir d'un critère de distance qui concerne toutes les données situées à proximité (dans la limite du rayon de la fenêtre de lissage).

Pour que des champs KDE puissent alimenter un SOLAP, nous avons émis l'hypothèse que pour chaque opération impliquant une agrégation de ces champs, l'utilisateur devrait avoir l'illusion de calculer un champ KDE sur base d'un nuage de points défini par les paramètres de la dite opération. L'outil SOLAP pourrait alors être considéré comme une interface rapide, conviviale et élégante pour la production de cartes de hotspots lorsqu'elle intègre des champs KDE (agréger des champs est globalement plus rapide que calculer des KDE). De plus, ces hotspots pourraient faire l'objet d'une analyse multidimensionnelle à travers les interfaces graphiques ou tabulaires. Nous avons ensuite démontré que pour que cette hypothèse soit vraie, une mesure ROLAP « champ KDE » doit être contrainte par deux métadonnées spatiales supplémentaires : la fonction KDE et la taille de la fenêtre de lissage.

A partir d'un exemple d'hypercube raster KDE intégrant des données criminelles de la ville de Londres, nous avons illustré plusieurs opérations SOLAP impliquant une mesure « densité de délits » (mesure KDE), mais aussi une mesure « nombre de délits ». Il est en effet important pour l'utilisateur de manipuler ces deux mesures afin d'alimenter l'interface cartographique par des hotspots, tout en ayant accès aux quantités de délits absolues à l'intérieur de ces hotspots via l'interface graphique (ou tabulaire). Parmi ces opérations, celles appliquées sur l'interface cartographique (affichant les dimensions spatiales) sont évidemment les plus intéressantes pour les champs KDE. Il est ainsi possible d'exploiter des coupes non-spatiales pour définir les données de la carte produite, des coupes spatiales pour isoler certaines zones d'intérêt, ou encore des comparaisons cartographiques efficaces. Le forage spatial sur champs KDE est particulièrement intéressant lorsqu'il est appliqué entre des niveaux spatiaux définis par une fenêtre de lissage et/ou une fonction KDE différente(s). En effet, une modification de ces paramètres entraîne une généralisation cartographique que l'utilisateur peut adapter en temps réel à son échelle d'analyse : un « drill down » spatial produit une carte où les hotspots sont plus petits et plus nombreux, et donc plus adaptée à une analyse plus locale (plus grande échelle), et *vice versa* pour le « roll up ». Ce changement d'échelle en temps réel dans un

¹⁶⁵ « Kernel Density Estimation ».

¹⁶⁶ « Modifiable Areal Unit Problem ».

SOLAP KDE constitue une solution potentielle¹⁶⁷ à l'ambiguïté du choix de la fonction KDE et de la fenêtre de lissage pour la production de carte de hotspots, en particulier dans le domaine du « crime mapping » (voir section 7.3.3).

Ce chapitre a ainsi clairement démontré l'intérêt d'un SOLAP raster pour l'analyse de données spatialement continues, mais aussi discrètes. La suite de cette thèse sera consacrée à la validation de ce modèle au moyen d'une architecture web complètement « open source », notamment sur un hypercube de données impliquant davantage de dimensions non-spatiales et de niveaux spatiaux.

¹⁶⁷ Rappelons néanmoins que tous les faits détaillés doivent être pré-calculés durant la phase d'intégration de données de l'hypercube KDE pour chaque niveau spatial impliquant des paramètres KDE différents (fonction et fenêtre de lissage). Cette phase d'intégration est détaillée dans la section 8.3.3.2.

PARTIE 3 : VALIDATION ET CONCLUSION

« La théorie, c'est quand on sait tout et que rien ne fonctionne. La pratique, c'est quand tout fonctionne et que personne ne sait pourquoi... »

Albert Einstein

Chapitre 8 : validation

8.1. Introduction

Dans le présent chapitre, le modèle SOLAP raster développé dans la partie précédente est validé par un prototype. Ce prototype a permis la modélisation de trois hypercubes raster dont les principales caractéristiques sont présentées et comparées dans la section 8.2 : deux hypercubes raster exploitent des données de criminalité discrètes de manière spatialement continue via l'approche KDE (« Seattle » et « London »), et un cube raster exploite des données de températures nativement continues sur la surface de la Lune (« Moon »).

La section 8.3 dresse une brève description du prototype et des fonctionnalités qu'il offre pour chaque élément de son architecture web : entrepôt de données géré par PostGIS, serveur SOLAP écrit en PHP et géré par un serveur web Apache et interface client écrite en HTML et JavaScript. Cette dernière partie, détaillant les fonctionnalités propres des interfaces cartographiques et graphiques du prototype, fait également office de tutoriel pour les lecteurs de cette thèse désirant tester l'outil en y accédant via l'adresse web suivante (un utilisateur et un mot de passe doivent être créés en cliquant sur le lien « create user » au-dessus à gauche de la page) :

<http://nolap01.ulg.ac.be/rastercube>

La section 8.4 illustre un type d'utilisation intéressante du SOLAP raster qui n'a pas encore réellement été abordée jusqu'ici : l'emploi de fonctions holistiques (étendue ou écart-type) pour les agrégations. Afin de mieux cerner l'utilité du raster dans le SOLAP de manière générale, une étude comparative est présentée dans la section 8.5. Celle-ci se base sur les temps de réponses de requêtes SOLAP identiques appliquées sur le même jeu de données discrètes selon trois approches différentes : SOLAP raster, SOLAP vectoriel et SIG. Les résultats de cette étude permettent notamment de définir les circonstances dans lesquelles un SOLAP raster peut compléter efficacement les fonctionnalités déjà offertes par les outils vectoriels. Avant de conclure ce chapitre, trois perspectives de recherche sont abordées dans le domaine des hypercubes raster (section 8.6) : les hypercubes raster tuilés, la distribution des requêtes sur plusieurs serveurs et l'exploitation de cubes raster dimensionnels.

8.2. Cas d'étude

Pour alimenter le prototype SOLAP raster présenté à la section 8.3, trois cas d'étude ont été traités. Les deux premiers sont des hypercubes raster KDE (voir chapitre 7) impliquant des données de criminalité. L'un concerne la ville de Londres et l'autre la ville de Seattle. Le troisième est un cube qui intègre directement des raster continus modélisant la température à la surface de la Lune. Les caractéristiques dimensionnelles propres à chacun de ces hypercubes entraînent un volume de données stockées plus ou moins important. Dans la comparaison dressée par le Tableau 3, on peut ainsi constater que l'hypercube cube « Seattle » qui compte le plus de dimensions (2 dimensions raster et 4

dimensions ROLAP) est logiquement le plus volumineux, avec environ 7 milliards de faits détaillés.

	London	Seattle	Moon
Nombre de mesures	2	2	2
Nombre de dimensions ROLAP	2	4	1
Nombres de dimensions raster	2	2	2
Nombre de niveaux hiérarchiques spatiaux	4	4	1
Nombre de faits détaillés ROLAP	132	11 304	100
Nombre de faits raster par fait ROLAP de l'hypercube de base	1500 colonnes * 1950 rangées	525 colonnes * 1194 rangées	1440 colonnes * 721 rangées
Nombre total de faits raster pour l'hypercube de base	Environ 386 millions	Environ 7 milliards	Environ 103 millions
Nombre de faits raster par fait ROLAP du cuboïde le moins détaillé	59 colonnes * 45 rangées	16 colonnes * 36 colonnes	1440 colonnes * 721 rangées
Nombre total de faits raster pour le cuboïde le moins détaillé	Environ 350 000	Environ 6 millions	Environ 103 millions

Tableau 3. Comparaison du volume de données des trois hypercubes raster

8.2.1. Hypercube London

Le premier hypercube KDE modélise la criminalité de la ville de Londres (UK Police, 2014) et a servi de base à tous les exemples fournis dans les trois chapitres précédents. Son schéma conceptuel est illustré dans la Figure 88. Puisqu'il est composé de deux dimensions ROLAP comptant respectivement 12 mois (année 2012) et 11 types de délits (membres détaillés), le nombre total de faits ROLAP est 132 (le cube présente une densité ROLAP de 100 %). Notons que contrairement aux exemples des chapitres précédents, le schéma compte en réalité 4 niveaux hiérarchiques spatiaux qui sont pré-calculés dans des cuboïdes matérialisés.

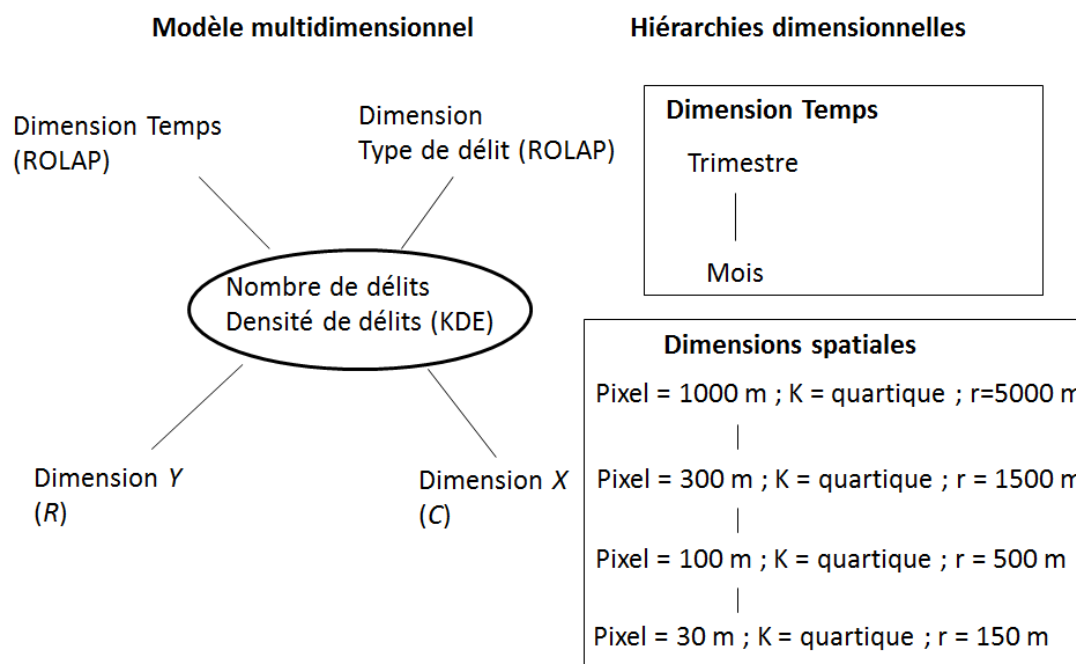


Figure 88. Schéma conceptuel de l'hypercube raster « London »

8.2.2. Hypercube Seattle

Dans le second hypercube raster KDE, les données de criminalité proviennent des appels au 911 reçus par la Police de Seattle au cours de l'année 2013 (Seattle Police, 2015). Par rapport à l'hypercube « London », les données de Seattle sont sémantiquement beaucoup plus précises puisque 4 dimensions ROLAP sont présentes dans le schéma multidimensionnel (Figure 89). Une dimension « type de délit » compte 25 membres détaillés, eux-mêmes hiérarchisés selon 4 groupes de délits, et le temps est décomposé en trois dimensions : le mois (12 membres détaillés avec un niveau supérieur « trimestre »), le jour de la semaine (7 membres détaillés) et la tranche de trois heures (8 membres détaillés).

Si l'hypercube présentait une densité ROLAP de 100%, le nombre total de faits détaillés devrait être de 16800 ($25 \times 12 \times 7 \times 8$). Puisque le nombre total de fait détaillés ROLAP stockés vaut en réalité 11304, la densité des dimensions ROLAP vaut environ 67% ($11304/16800$). Tous les faits détaillés théoriquement possibles ne sont donc pas systématiquement représentés dans les données (par exemple, le type de délit « homicide » ne se retrouve pas dans chaque combinaison possible de tranche horaire/jour de la semaine/mois). Cette économie d'espace de 33 % est un avantage des architectures ROLAP expliqué à la section 1.3.5.1. En revanche, puisque les dimensions raster sont assimilables à des dimensions MOLAP, les problèmes de gestion de la faible densité selon les dimensions spatiales impliquent une surconsommation d'espace des mesures ROLAP (raster) ne pouvant être résolue que par des méthodes de compression¹⁶⁸ (voir section 1.3.5.2) ou par le tuilage des raster (voir section 8.6.1).

¹⁶⁸ A ce jour, PostGIS raster ne propose hélas aucune méthode de compression.

Enfin, les quatre niveaux hiérarchiques spatiaux matérialisés par des cuboïdes (mesure ROLAP) sont exactement les mêmes que pour le cube London, tout comme les mesures « nombre de délits » (par pixel) et « densité de délits » (KDE).

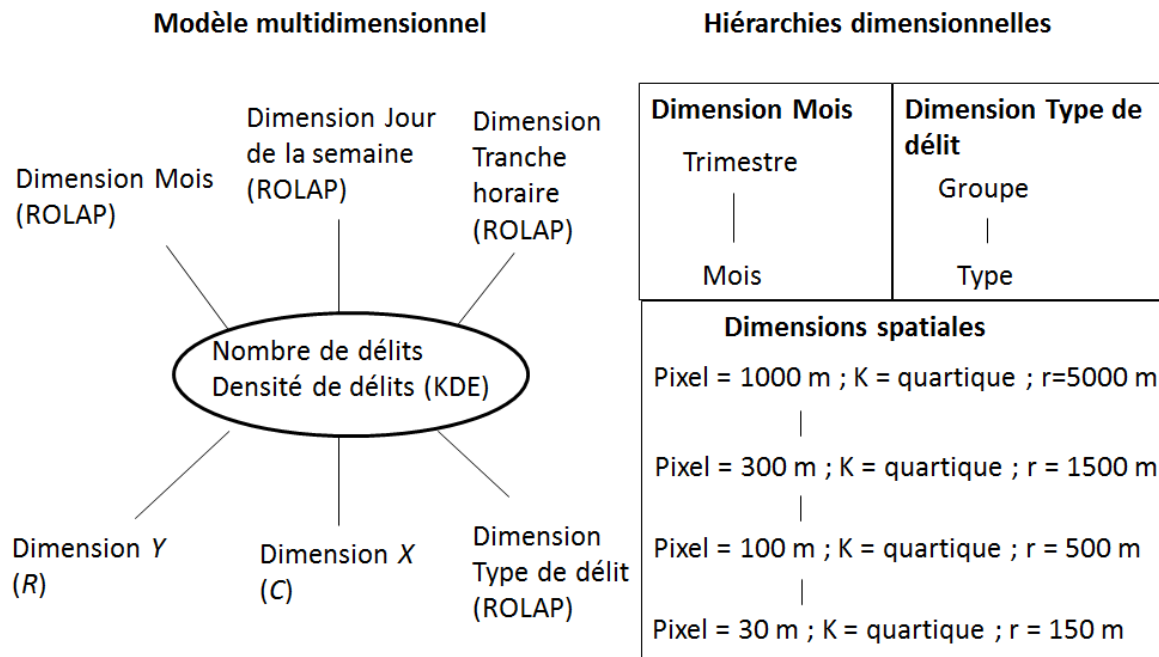


Figure 89. Schéma conceptuel de l'hypercube raster « Seattle »

8.2.3. Cube Moon

En plus de son originalité, le dernier cube raster, qui intègre des données de température de la surface de la Lune, démontre que notre prototype est également capable de gérer des données nativement continues (et non pas uniquement des KDE). Les données proviennent du capteur Diviner à bord du « Lunar Reconnaissance Orbiter » de la NASA (Paige *et al*, 2011). Le cube raster correspondant présente deux dimensions spatiales : la latitude (dimension raster *R*) et la longitude (dimension raster *C*). Le seul niveau spatial modélisé est caractérisé par une taille de pixel de 0,25 ° (soit environ 7,6 km le long de l'équateur puisque la Lune a un rayon de 1737 km). Notons que comme pour tout raster « non-projeté » (c'est-à-dire géoréférencé uniquement par rapport à une sphère de référence), les données résultent en réalité d'une projection cylindrique « plate-carrée » (Donnay, 2010).

Ce schéma comprend également une dimension temporelle qui compte 100 membres détaillés représentant chacun 1/100 de jour lunaire. Le niveau hiérarchique supérieur de cette dimension représente 1/10 de jour lunaire. Puisque les données intégrées résultent de plusieurs observations, les deux mesures correspondent respectivement à une moyenne ou un écart-type de la température.

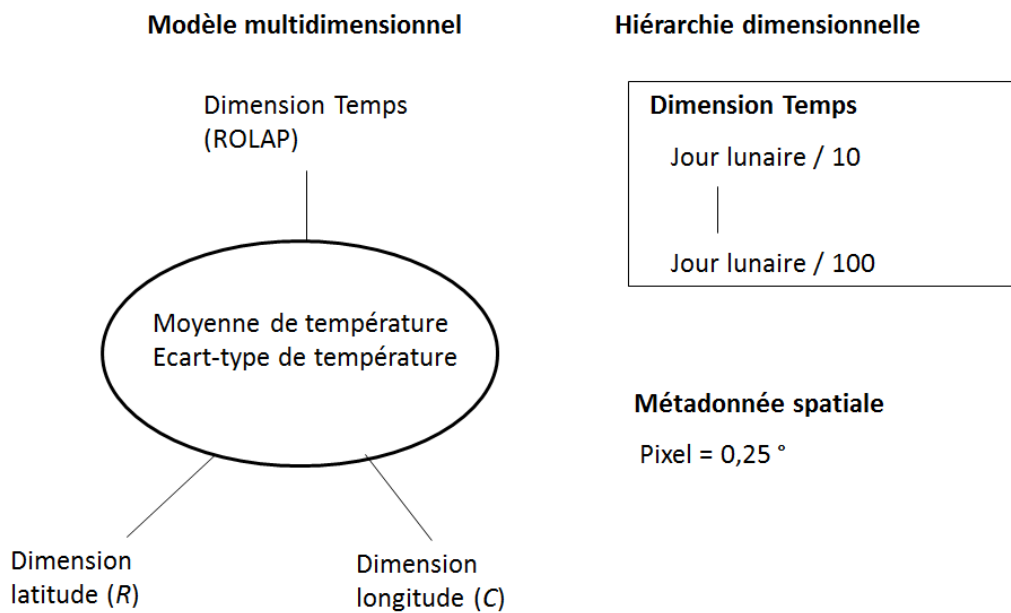


Figure 90. Schéma conceptuel du cube raster « Moon »

La Figure 91 est un exemple de coupe temporelle sur le cube Moon dans l'interface du prototype SOLAP raster présentée à la section 8.3.4. Notons que l'échelle graphique n'est pas adaptée puisque PostGIS (SGBD de l'entrepôt de données) ne gère aucun SCR lunaire. Le SCR employé ici est donc un système terrestre (EPSG : 4326). Pour être correcte, cette échelle devrait être multipliée par le rapport rayon lunaire/rayon terrestre (nous ne tenons pas compte non plus des altérations d'échelle dues à la projection cylindrique « plate-carrée » lorsque l'on se rapproche des pôles).

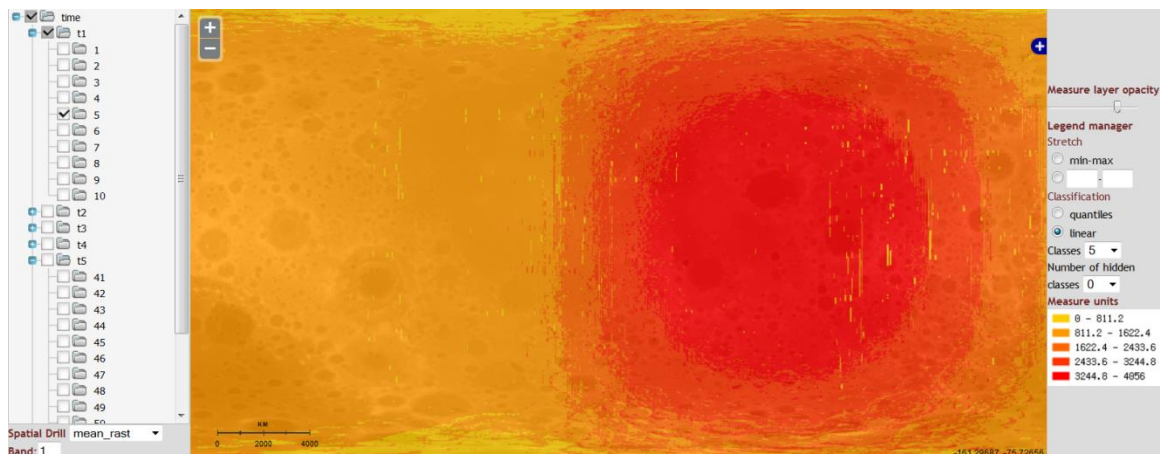


Figure 91. Exemple de coupe temporelle sur le cube « Moon »

8.3. Prototype SOLAP raster

Cette importante section est consacrée au prototype qui valide notre modèle SOLAP raster. Celui-ci est d'abord décrit de manière globale (section 8.3.1), puis de manière détaillée à travers les trois principaux éléments de son architecture : l'entrepôt de données (section 8.3.2), le serveur SOLAP (section 8.3.3) et l'interface client (section 8.3.4). En particulier, cette dernière partie donne suffisamment d'informations pour que les lecteurs de cette thèse puissent eux-mêmes tester l'interface.

8.3.1. Architecture globale

L'architecture client-serveur du prototype SOLAP raster est uniquement composée d'outils « open source » (Figure 92). Elle est inspirée des architectures BI classiques décrites au chapitre 1. Côté serveur, des données externes et hétérogènes sont intégrées par l'ETL GeoKettle¹⁶⁹ (Spatialytics, 2014) dans un entrepôt de données¹⁷⁰ géré par le SGBD PostgreSQL et son extension PostGIS raster (Racine & Cumming, 2011). Cette partie sera détaillée dans la section 8.3.2.

En aval de l'entrepôt, un serveur SOLAP programmé en PHP interprète la structure multidimensionnelle de l'entrepôt relationnel (architecture ROLAP) et construit ainsi les requêtes SQL pour l'interrogation de l'hypercube raster. Les résultats de ces requêtes sont renvoyés directement au client lorsqu'il s'agit de graphiques. Lorsque les résultats sont cartographiques, ceux-ci sont transmis sous forme de WMS¹⁷¹ (ISO, 2005) par le serveur de données spatiales MapServer (2014). Toute cette partie est gérée par le serveur web Apache (2015). Comme nous le verrons dans la section 8.3.3.2, le serveur SOLAP est également capable de construire automatiquement un hypercube raster KDE sur base d'un entrepôt de données vectoriel (nuages de points).

Côté client, les interactions avec le serveur SOLAP et l'affichage des résultats ont lieu à travers les langages HTML et JavaScript (n'importe quel navigateur peut donc supporter l'interface client). En particulier, la librairie OpenLayers (2015) fournit toutes les fonctionnalités de base de l'interface cartographique (import des couches WMS, outils de navigation, digitalisation, etc.) et la librairie DHTMLX (2015) est exploitée pour la gestion des graphiques et d'un arbre à dimensions permettant le paramétrage des opérations SOLAP sur l'hypercube. Les fonctionnalités offertes par l'interface seront passées en revue dans la section 8.3.4.

Notons que pour assurer une bonne fluidité des opérations client-serveur (et *vice versa*), la technologie AJAX (« Asynchronous JavaScript and XML ») permet au client JavaScript d'interroger le serveur PHP de manière asynchrone sans nécessiter le rechargement complet de la page web à chaque opération.

¹⁶⁹ Geokettle n'étant pas (encore) capable de gérer les données raster, l'intégration de raster bruts a été effectuée en exploitant directement la fonction raster2pgsql fournie par PostGIS.

¹⁷⁰ Notons que l'entrepôt de données est également capable de gérer le mode vectoriel et pourrait éventuellement être exploité à la fois par un serveur SOLAP raster et vectoriel (voir section 8.3.3).

¹⁷¹ « Web Map Service ».

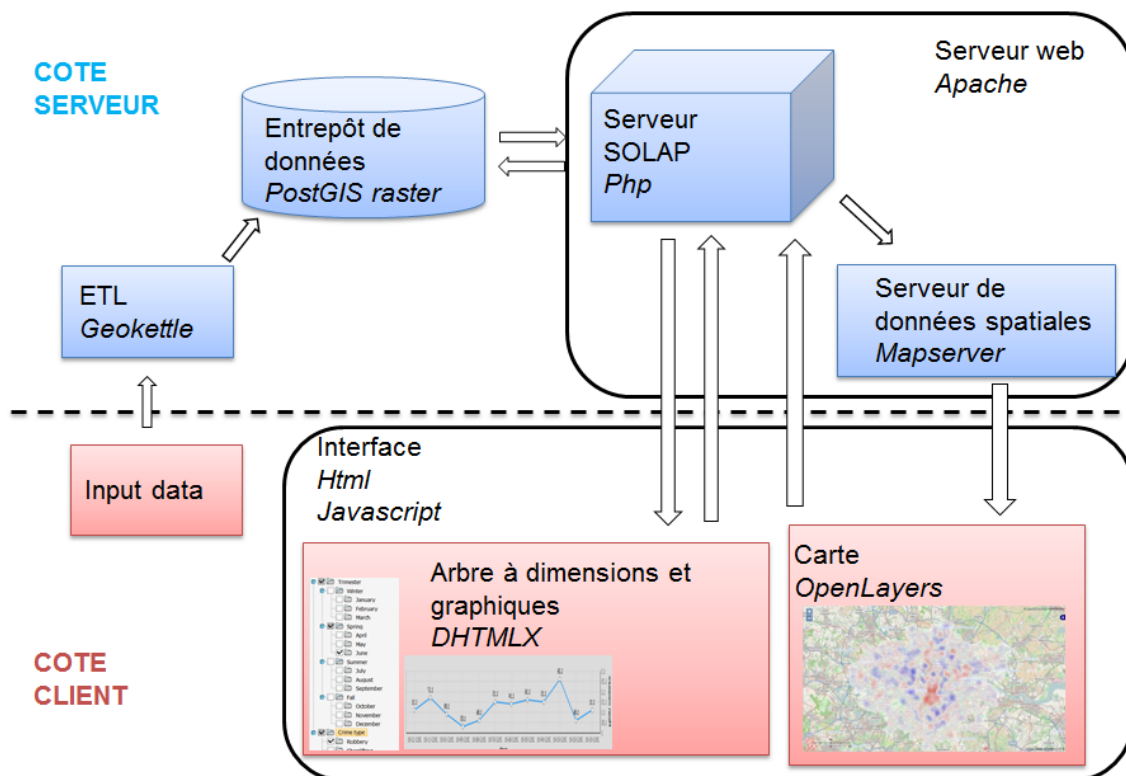


Figure 92. Architecture du prototype SOLAP raster

8.3.2. Entrepôt de données

Comme déjà expliqué, l'entrepôt de données est le cœur d'une architecture BI. Celui-ci stocke aussi bien les données détaillées (faits détaillés) que les données pré-agrégées (cuboïdes matérialisés) qui seront exploitées par le serveur OLAP. Dans notre prototype SOLAP raster, cet entrepôt est géré par le SGBD relationnel PostGreSQL et son extension PostGIS. Il s'agit donc d'une architecture ROLAP. Grâce à PostGIS, il est possible de considérer les mesures ROLAP comme des tableaux raster comptant jusqu'à trois dimensions : *R*, *C* et *B*. Notre architecture présente donc également des caractéristiques similaires au MOLAP puisque ces tableaux raster présentent une indexation implicite en accord avec la structure multidimensionnelle des données pour les dimensions raster. Pour agréger ces raster, les fonctions d'algèbre de carte proposées par PostGIS ont été exploitées.

8.3.2.1. Modèle logique

La Figure 93 représente le modèle logique de l'entrepôt de données exploité par le prototype SOLAP raster. Pour les besoins de ce travail, les schémas en étoile des trois jeux de données décrits à la section 8.2 ont été implémentés de la manière la plus simple : une table des faits dégénérée comprenant à la fois les mesures et les membres de dimension des faits ROLAP détaillés. Ainsi, les tables « fact_raster », « seattle_fact_table » et « moon_fact2 » représentent à elles seules les hypercubes raster respectifs « London », « Seattle » et « Moon ». Comme nous l'avons vu, cette technique est déconseillée dans les

systèmes ROLAP en raison du très grand nombre d'enregistrements (faits détaillés) généralement présents dans une table des faits (Kimball & Ross, 2013). Il est alors préférable d'utiliser un schéma en étoile ou en flocon (voir section 1.2.4) afin de normaliser les dimensions et gagner un espace de stockage considérable dans la table des faits. Cependant, dans notre modèle, une grande partie des faits détaillés est définie au niveau des mesures ROLAP (dimensions raster), si bien que le nombre d'enregistrements de la table des faits ROLAP est d'environ 10 000 pour l'hypercube le plus volumineux (Seattle), et d'environ une centaine pour les deux autres hypercubes. Nous sommes donc loin des entrepôts comptant des millions d'enregistrements relationnels et estimons qu'un schéma en étoile dégénéré est suffisant pour valider le prototype.

En ce qui concerne les hypercubes raster KDE, une table des faits vectorielle stocke également les données individuelles : « crime_point2 » pour l'hypercube « London » et « seattle_crime_point » pour l'hypercube « Seattle ». Dans ces tables, chaque délit occupe un enregistrement et est défini au niveau le plus détaillé de ses dimensions (spatiales et non-spatiales). Du point de vue spatial, chaque enregistrement est défini par un point dans le système de coordonnées de référence adéquat (« British National Grid¹⁷² » pour London et « Washington North¹⁷³ » pour Seattle). Les hypercubes vectoriels comptent ainsi 1 130 646 délits pour Londres et 806 663 délits pour Seattle. Bien que cet aspect ne soit pas implémenté dans ce prototype, ces tables des faits vectorielles auraient pu également servir de base à un SOLAP vectoriel couplé au SOLAP raster. Dans ce travail, ces tables ont été exploitées uniquement pour construire les hypercubes KDE correspondant (« fact_raster » et « seattle_fact_table ») et pour effectuer les tests de comparaison vecteur/raster de la section 8.5. Aussi bien pour Londres que pour Seattle, les données ont été fournies sous forme de fichier Excel comptant une colonne « latitude » et une colonne « longitude » pour la localisation des délits. La précision de ces délits est au niveau de la rue.

Pour Londres et Seattle, la densité de délits est une mesure qui occupe la bande 1 des raster, et le nombre de délits par pixel occupe la bande 2. Puisque dans cette modélisation, la bande est considérée comme une mesure différente et non comme une dimension (*B*), aucune opération d'agrégation sur plusieurs bandes ne peut avoir lieu et chaque mesure ROLAP comprend uniquement les deux dimensions *R* et *C*. Dans le cube raster « Moon », toutes les mesures occupent uniquement la bande 1. Ainsi, l'exploitation de la dimension *B*, notamment pour modéliser une dimension continue, n'est pas validée dans ce prototype. Cet aspect de notre modélisation aura donc été décrit uniquement d'un point de vue théorique dans le cadre de la thèse. L'implémentation de la dimension raster *B* pourrait éventuellement faire l'objet de futures recherches, notamment avec d'autres SGBD (Oracle, par exemple).

Afin de gérer l'exploitation de l'entrepôt simultanément par plusieurs utilisateurs, la table « user_list » stocke les informations sur ces derniers. Chaque raster calculé par une opération SOLAP est donc stocké dans la table « map_raster » avec le nom de l'utilisateur ayant effectué l'opération. Ce stockage physique des résultats dans la base de données était nécessaire afin de ne pas systématiquement recalculer la requête à chaque rechargement du WMS (qui a lieu pour chaque opération de navigation dans l'interface cartographique gérée par OpenLayers). Il est également utile de conserver l'historique des opérations d'un

¹⁷² EPSG : 27700.

¹⁷³ EPSG : 2855.

utilisateur pour gérer les opérations SOLAP de comparaison (voir section 5.9.3). Cet historique est effacé à chaque nouvelle connexion de l'utilisateur. Précisons que seuls les résultats des requêtes sur l'interface cartographique sont stockés dans la base de données. Les résultats de l'interface graphique, quant à eux, sont simplement mémorisés dans le cache du navigateur du client.

Enfin, la table « geodimension » mémorise les entités vectorielles digitalisées à la volée par l'utilisateur. Ces entités peuvent être considérées comme des membres spatialisés lorsque l'utilisateur désire les intégrer dans son analyse. Elles sont alors simplement rasterisées pour être intégrées à la structure multidimensionnelle de l'entrepôt lors d'une opération impliquant un niveau spatial donné (par exemple : « pixel = 300 m »).

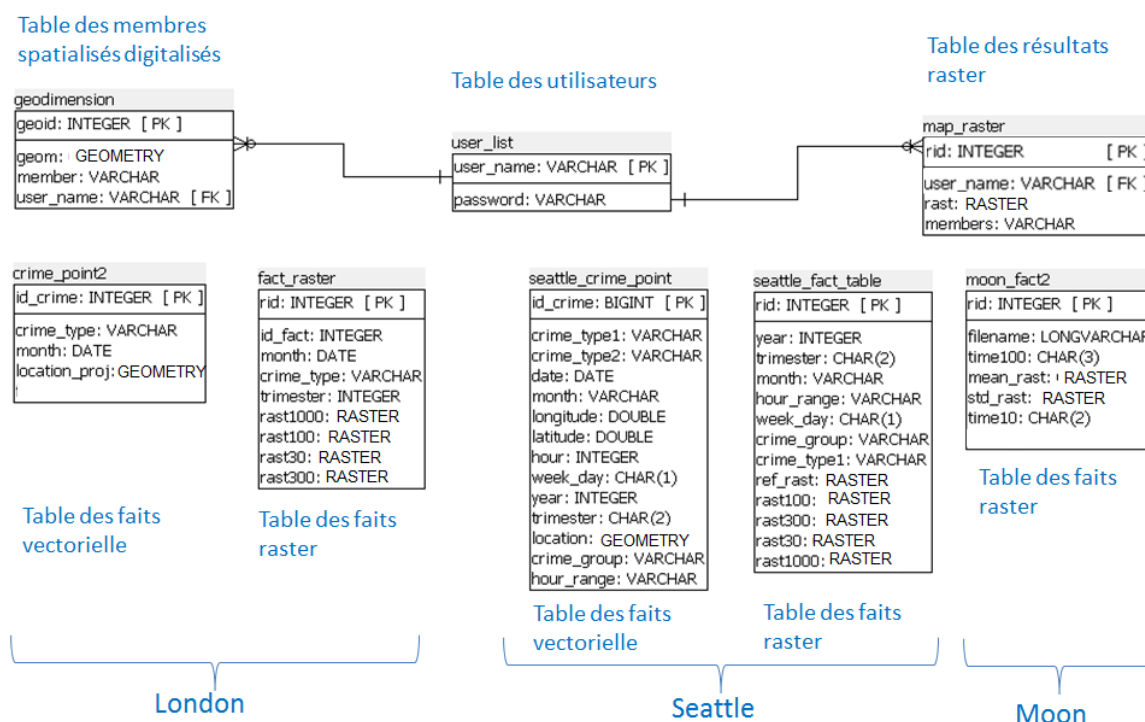


Figure 93. Tables principales de l'entrepôt de données SOLAP raster

8.3.2.2. Métadonnées

Dans la section 6.4, nous avons vu que pour optimiser les opérations SOLAP d'un hypercube raster, il est nécessaire de contraindre les mesures ROLAP (raster ou cubes raster) au moyen de plusieurs paramètres : les métadonnées spatiales et dimensionnelles. PostGIS permet de fixer ces métadonnées par le biais de la vue « raster_columns » (voir section 2.4.2.2). Les informations de la table de métadonnées théorique présentée dans la Figure 72 sont ainsi toutes définies dans cette vue. A titre d'exemple, la Figure 94 représente les contraintes de la vue « raster_columns » pour l'hypercube London :

- `r_table_catalog` : nom de l'entrepôt
- `t_table_name` : nom de la table des faits
- `r_table_column` : nom de la mesure ROLAP
- `srid` : code EPSG du système de coordonnées de référence (métadonnée spatiale SCR)
- `scale_x` et `scale_y` : paramètres d'homothétie h_C et h_R (métadonnées spatiales)
- `blocksize_x` et `blocksize_y` : métadonnées dimensionnelles N_C et N_R
- `same_alignment` : fixe les mêmes métadonnées spatiales t_C et t_R si ce paramètre est défini sur « TRUE »
- `num_bands` : métadonnée dimensionnelle N_B (ici, la bande est exploitée comme mesure et non comme dimension)
- Autres paramètres :
 - o `regular_blocking` : fixe la taille des tuiles (non exploité ici)
 - o `pixel_types` : codage des valeurs des pixels (mesures raster)
 - o `nodata_values` : valeur « no data » éventuelle
 - o `out_db` : possibilité de stocker le raster sous forme de fichier en dehors de l'entrepôt (inexploité ici)
 - o `extent` : enveloppe convexe de la mesure ROLAP (polygone vectoriel)

	<code>r_table_catalog</code> name	<code>r_table_schema</code> name	<code>r_table_name</code> name	<code>r_raster_column</code> name	<code>srid</code> integer	<code>scale_x</code> double prec	<code>scale_y</code> double prec	<code>blocksize_x</code> integer	<code>blocksize_y</code> integer
1	backup	public	fact raster	rast1000	27700	1000	-1000	59	45
2	backup	public	fact raster	rast100	27700	100	-100	585	450
3	backup	public	fact raster	rast30	27700	30	-30	1950	1500
4	backup	public	fact raster	rast300	27700	300	-300	195	150

	<code>same_alignment</code> boolean	<code>regular_blocking</code> boolean	<code>num_bands</code> integer	<code>pixel_types</code> text[]	<code>nodata_values</code> double precis	<code>out_db</code> boolean[]	<code>extent</code> geometry
1	TRUE	FALSE	2	{32BF, 32F}		{f, f}	0103000002
2	TRUE	FALSE	2	{32BF, 32F}		{f, f}	0103000002
3	TRUE	FALSE	2	{32BF, 32F}		{f, f}	0103000002
4	TRUE	FALSE	2	{32BF, 32F}		{f, f}	0103000002

Figure 94. Extrait de la vue de métadonnées « `raster_columns` » pour les mesures de l'hypercube « London »

Notons que les métadonnées spécifiques aux mesures KDE (fonction KDE et fenêtre de lissage), quant à elles, ne sont pas prises en charge par PostGIS. Elles doivent donc être gérées au niveau du serveur SOLAP (voir section 8.3.3).

8.3.2.3. Requêtes SOLAP SQL

PostGIS intègre diverses fonctions, notamment d'algèbre de carte, manipulant des raster directement dans sa syntaxe SQL. Puisque, comme nous l'avons vu lors des chapitres 5 et 6, l'algèbre de carte est très souvent exploitée par les opérations élémentaires d'agrégation dans un hypercube raster, il est possible de construire les requêtes SOLAP en exploitant uniquement cette syntaxe SQL. A titre d'exemples, la Figure 95 et la Figure 96 illustrent quelques requêtes SOLAP typiques sur l'hypercube Seattle. Puisque cette section n'a pas pour but de rentrer dans les détails du SQL de PostGIS, nous renvoyons au manuel du SGBD pour plus d'informations sur le sujet (PostGIS, 2015).

Dans la Figure 95a, la mesure « densité de délits » (bande 1) est agrégée non-spatialement et est filtrée sur le membre « transportation » de la dimension « type de délit » (niveau « crime_group »). Le résultat est une carte affichant les dimensions spatiales X, Y, et la requête SQL implique donc un filtrage ROLAP suivi d'une agrégation totale ROLAP qui correspond à une opération locale d'algèbre de carte. Cette dernière opération est appliquée à l'aide de la fonction ST_Union.

Dans la Figure 95b, une requête similaire est appliquée en faisant intervenir une coupe spatiale. Pour paramétrer cette coupe, un membre spatialisé « north » a été directement digitalisé par l'utilisateur « jpk », puis stocké dans la table geodimension. Avant de procéder à la requête de la Figure 95b, ce membre a été préalablement rasterisé¹⁷⁴ dans une vue « maskjpk¹⁷⁵ » dont le rôle est la définition des filtres spatiaux intervenant dans les requêtes SOLAP. Le raster défini dans cette vue est donc un masque binaire partageant les mêmes métadonnées spatiales que la mesure ROLAP agrégée : « rast300 » (résolution spatiale de 300 m par pixel). Au moment de l'opération SOLAP, ce masque est multiplié par l'hypercube raster afin d'isoler les pixels du membre spatialisé « north » grâce à la fonction ST_MapAlgebraExpr. Ce filtrage raster (spatial) intervient directement après le filtrage ROLAP (non-spatial).

En raison du manque de temps et des difficultés techniques concernant l'implémentation de fonctions zonales d'algèbre de carte au moyen de PostGIS, l'affichage de membres spatialisés dans l'interface cartographique n'a pas pu être intégré au prototype. Comme expliqué à la section 5.8.3, cet aspect a été validé en exploitant les résultats stockés dans la table « map_raster » (qui stocke tous les résultats des requêtes SOLAP d'un utilisateur) à l'aide du SIG-logiciel QGIS (l'agrégation zonale est appliquée en aval de la chaîne d'opérations élémentaires). L'affichage de membres spatialisés dans l'interface graphique, quant à lui, est bien validé par le prototype (section 8.3.4.2).

¹⁷⁴ Au moyen de la fonction ST_Asraster.

¹⁷⁵ Une vue personnalisée est créée pour chaque utilisateur afin d'y stocker les requêtes construisant les filtres spatiaux (gérés comme des masques binaires).

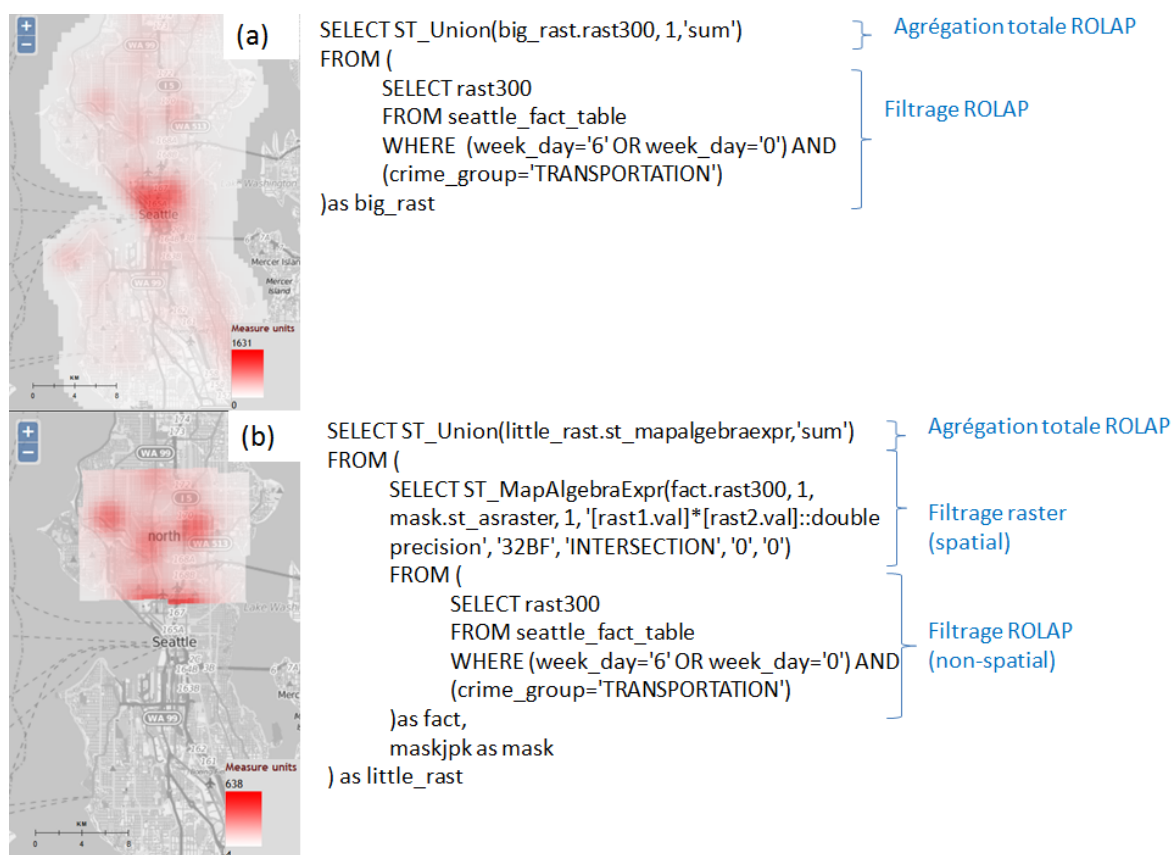
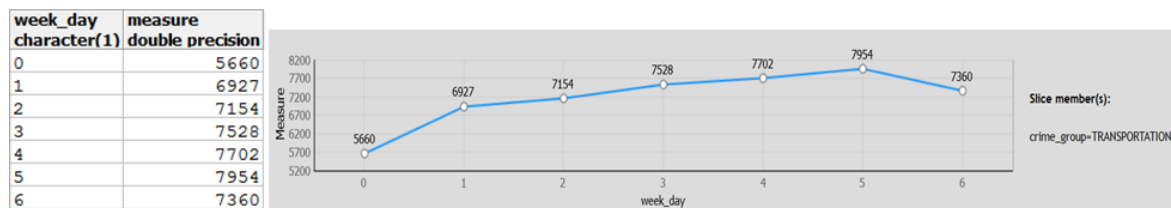


Figure 95. Exemples de requêtes SOLAP SQL dans l'interface cartographique

Dans la Figure 96, un graphique du nombre de délits par jour de la semaine¹⁷⁶ est construit sur base de la coupe de la Figure 95b. Puisqu'il est cette fois question de l'interface graphique, la dimension affichée est une dimension ROLAP non-spatiale et la mesure exploitée est le nombre de délits (bande 2). Dans l'ordre, la requête SQL implique un filtrage ROLAP (membre « transportation » de la dimension « type de délit »), un filtrage raster (coupe spatiale sur le membre spatialisé « north »), une agrégation totale ROLAP sur les dimensions ROLAP non-affichées (impliquant donc un « GROUP BY » sur les membres de la dimension « week_day »), et une agrégation totale raster au moyen de la fonction ST_SummaryStats (agrégation spatiale de tous les pixels).

¹⁷⁶ Dans l'exemple de la Figure 96, 0 correspond à dimanche, 1 correspond à lundi, etc.



```

SELECT week_day, (stats).sum as measure
FROM (
  SELECT ST_SummaryStats(rast) As stats, week_day
  FROM (
    SELECT ST_Union(little_rast.st_mapalgebraexpr,'sum') as rast, week_day
    FROM (
      SELECT ST_MapAlgebraExpr(fact.rast300, 2, mask.st_asraster, 1,
        '[rast1.val]*[rast2.val]::double precision', '32BF', 'INTERSECTION', '0',
        '0'), week_day
      FROM (
        select rast300, week_day from seattle_fact_table
        WHERE (crime_group='TRANSPORTATION')
      ) as fact,
      maskjpk as mask
    ) as little_rast
    GROUP BY week_day ORDER BY week_day
  ) as foo
) as stats

```

Agrégation totale raster (spatiale)
 Agrégation totale ROLAP (non-spatiale)
 Filtrage raster (spatial)
 Filtrage ROLAP (non-spatial)

Figure 96. Exemple de requête SOLAP SQL dans l'interface graphique

Notons que pour afficher plus d'une dimension dans l'interface graphique, le serveur SOLAP (voir section 8.3.3) calcule une boucle d'autant d'itérations (requêtes SQL) que de membres à afficher dans la dimension « inter graphiques ».

8.3.3. Serveur SOLAP

Le rôle principal du serveur SOLAP de ce prototype est l'interprétation de la structure multidimensionnelle des hypercubes raster. A partir des paramètres sélectionnés dans l'interface par l'utilisateur, celui-ci construit les requêtes SOLAP SQL transmises à l'entrepôt de données en se basant sur un fichier XML de définition de l'hypercube et sur les membres spatialisés stockés dans la table geodimension. Lorsque les résultats sont cartographiques, le serveur adapte également la classification de la carte en fonction du résultat de la requête et des paramètres choisis par l'utilisateur. Cet aspect sera détaillé dans la section 8.3.4.1.3.

En amont de l'analyse, le serveur SOLAP est également capable de construire un hypercube raster KDE sur base d'un entrepôt de données vectorielles (nuages de points). Cette partie sera détaillée dans la section 8.3.3.2.

8.3.3.1. Gestion des hypercubes raster et construction des requêtes SOLAP

Pour que la structure multidimensionnelle ROLAP de l'entrepôt puisse être interprétée comme telle par le serveur SOLAP, un fichier XML de définition d'un hypercube doit être choisi par l'utilisateur. Le module permettant la sélection d'un

hypercube parmi ceux stockés dans l'entrepôt est accessible via le menu « change data cube » de l'interface client (voir section 8.3.4).

Les trois fichiers XML correspondant aux hypercubes présentés à la section 8.2 sont appelés respectivement « london_ok.xml », « seattle_ok.xml » et « moon.xml ». Ces fichiers, détaillés dans l'annexe 1, contiennent les informations suivantes :

- Le nom de la table des faits raster
- Le nom de la table des faits vectorielle et de l'attribut spatial de ces faits (uniquement pour les cubes raster KDE)
- La définition hiérarchique de chaque dimension, niveau et membre présentée comme un arbre (les identifiants des membres et niveaux sont exactement les mêmes que ceux de l'entrepôt)

Remarquons que dans notre prototype, chaque membre est défini individuellement dans le fichier XML alors qu'il aurait suffi de définir les hiérarchies dimensionnelles uniquement sur base des noms des attributs correspondant aux différents niveaux dans l'ED (et des tables qui les comprennent dans le cas d'un schéma logique en étoile ou en flocon), comme c'est par exemple le cas avec le serveur SOLAP Mondrian (Pentaho, 2014). Bien que plus fastidieuse, la technique utilisée ici offre la possibilité de construire un hypercube KDE (voir section suivante) en omettant certains membres de l'entrepôt vectoriel sans devoir physiquement supprimer ces membres dans l'ED vectoriel (par exemple : construction d'un cube sur une période d'un mois au lieu d'un an). De plus, ce fichier XML est également utilisé pour la gestion de l'arbre à dimensions de l'interface client (voir section 8.3.4.1.1)

Lorsqu'une requête SOLAP est transmise depuis l'interface client vers le serveur SOLAP, celui-ci construit la requête SQL correspondante pour l'entrepôt de données. Les agrégations et filtres des dimensions ROLAP sont interprétés grâce au fichier XML de définition de l'hypercube et les agrégations et filtres sur les dimensions spatiales (raster) sont résolus par les opérations raster implémentées dans PostGIS.

Le serveur SOLAP transmet également les membres spatialisés digitalisés par l'utilisateur à l'entrepôt qui les stocke au format vectoriel. Lorsqu'une requête SOLAP fait intervenir un ou plusieurs membres spatialisés (filtrage spatial ou affichage de membres spatialisés dans l'interface), le serveur SOLAP construit la requête permettant la rasterisation des membres spatialisés sous forme de masque binaire.

De plus, lorsqu'une requête nécessite un affichage cartographique, le serveur SOLAP édite le fichier « map » qui définit le WMS (« Web Map Service ») reçu par le client (interprété par le serveur de données spatiales MapServer). Ce WMS est spécifique pour chaque utilisateur et va rechercher le dernier résultat calculé pour l'utilisateur en question dans la table « map_raster ». Un exemple d'un tel fichier « map » est fourni dans l'annexe 2. C'est notamment dans ce fichier que sont définis les paramètres de la symbolisation de la carte, et plus spécifiquement la classification des pixels du raster résultant de la requête SOLAP (étirement linéaire, classification par quantiles, etc.).

8.3.3.2. Construction d'hypercubes raster KDE

En plus de la construction des requêtes lors de l'analyse, le serveur SOLAP est également capable de créer un hypercube raster KDE sur base d'un entrepôt de données vectoriel stockant un nuage de points (phase d'intégration des données). Comme le montre la Figure 97, les hypercubes raster KDE construits de cette manière peuvent être considérés comme des magasins de données semblables à ceux retrouvés dans une architecture hybride MOLAP/ROLAP (voir section 1.3.5.3). Puisque le raster présente des caractéristiques similaires au MOLAP, les magasins de données raster KDE ont le même rôle que les magasins de données MOLAP dans une architecture hybride : analyser rapidement de petits jeux de données (dimensionnalité réduite) issus de l'entrepôt vecteur tout en offrant les avantages de la visualisation SOLAP par KDE (cartes de hotspots). Notons que les modules présentés dans la suite de cette section sont uniquement accessibles par un utilisateur ayant les droits d'administrateur.

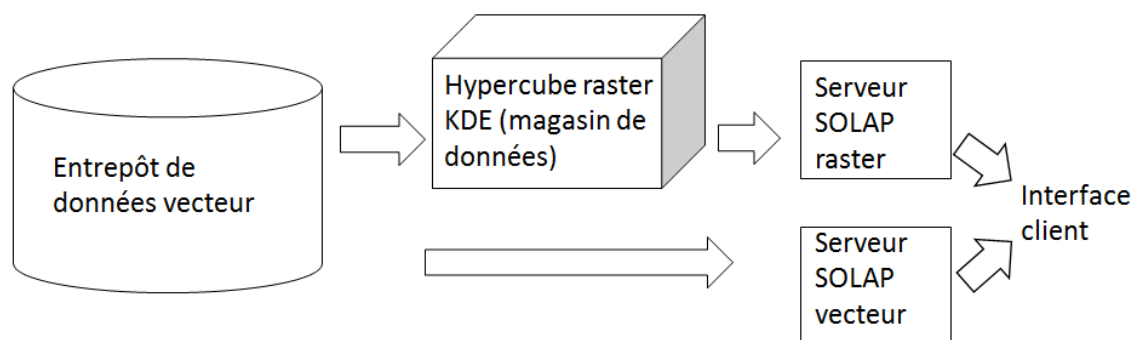


Figure 97. Architecture hybride exploitant un entrepôt de données vectoriel et des magasins de données raster KDE

Pour construire un hypercube KDE, la première étape est l'importation du fichier XML de définition de l'hypercube raster (Figure 98a). Grâce aux informations fournies dans le fichier, le serveur SOLAP crée la table des faits raster correspondante en allant rechercher les faits détaillés issus du fichier XML qui sont présents dans la table des faits de l'entrepôt vectoriel (il peut y avoir moins de faits détaillés que prévus théoriquement si l'hypercube présente une densité ROLAP de moins de 100%). Par exemple, pour l'hypercube London, la requête SQL de création de la table des faits raster construite par le serveur SOLAP est la suivante :

```
CREATE TABLE fact_raster AS (
SELECT DISTINCT month, crime_type FROM crime_point2
WHERE
(month='2012-01-01' OR month='2012-02-01' OR month='2012-03-01' OR month='2012-
04-01' OR month='2012-05-01' OR month='2012-06-01' OR month='2012-07-01' OR
month='2012-08-01' OR month='2012-09-01' OR month='2012-10-01' OR month='2012-
11-01' OR month='2012-12-01')
AND
(crime_type='Robbery' OR crime_type='Shoplifting' OR crime_type='Criminal damage
and arson' OR crime_type='Burglary' OR crime_type='Public disorder and weapons' OR
```

```

crime_type='Drugs' OR crime_type='Violent crime' OR crime_type='Other theft' OR
crime_type='Vehicle crime' OR crime_type='Anti-social behaviour' OR
crime_type='Other crime')
)

```

Le résultat de cette requête est une table contenant un enregistrement pour chacun des 132 faits ROLAP détaillés résultant du produit cartésien des 11 types de délits par les 12 mois de l'année (l'hypercube London a une densité ROLAP de 100 %). Ces faits sont définis uniquement par leurs dimensions, et pas encore au niveau de leurs mesures. Notons qu'à cette étape, un raster de référence est associé à la table uniquement pour connaître l'emprise spatiale des futures mesures associées au cube. Celui-ci est un raster vide dont l'emprise spatiale est calculée à partir de l'enveloppe convexe du nuage de points stocké dans l'entrepôt vecteur.

L'étape suivante est la création des mesures ROLAP KDE (Figure 98b). PostGIS ne fournissant pas de fonctions d'estimation de la densité par la méthode des noyaux (KDE), celle-ci a été créée¹⁷⁷ pour les besoins de ce travail au moyen du langage procédural PLPGSQL. Les détails du code de création de cette fonction, ainsi qu'un exemple d'utilisation, sont fournis dans l'annexe 3.

Comme le montre la Figure 98b, la création d'une mesure KDE nécessite les paramètres suivants :

- Le nom de la mesure
- La précision des données¹⁷⁸
- La taille des pixels en X et en $-Y$ ($h_C = -h_R$)
- La taille de la fenêtre de lissage (r)
- Le raster de référence calculé à l'étape précédente afin d'en copier les métadonnées spatiales de translation (t_R, t_C) et le SCR, et afin de calculer les métadonnées dimensionnelles N_R (nombre de rangées) et N_C (nombre de colonnes) pour la nouvelle mesure KDE

Connaissant ainsi toutes les métadonnées spatiales, dimensionnelles et KDE¹⁷⁹ des raster associés à cette nouvelle mesure, pour chaque fait détaillé défini dans la table des faits raster, un raster est calculé par la fonction KDE sur base du nuage de points défini par le fait correspondant dans l'entrepôt vectoriel. En reprenant l'exemple de l'hypercube London, un KDE serait donc calculé pour les cambriolages de janvier, puis pour les cambriolages de février, etc. L'utilisateur est ensuite libre de créer d'autres mesures KDE correspondant à différents niveaux spatiaux matérialisés (cuboïdes). Rappelons qu'un niveau spatial (mesure ROLAP) est associé à une résolution et une fenêtre de lissage spécifique.

¹⁷⁷ Cette fonction a d'ailleurs fait l'objet d'un ticket sur : <http://trac.osgeo.org/postgis/ticket/2894>

¹⁷⁸ Afin d'accélérer l'exécution de la fonction PostGIS KDE, le nuage de points est d'abord transformé en grille vectorielle dont la résolution correspond à la précision des données. La valeur d'une maille de la grille est le nombre de points tombant dans cette maille. La fonction KDE est ensuite appliquée sur cette grille en tenant compte de la valeur des points dans sa pondération.

¹⁷⁹ La fonction PostGIS KDE est toujours quartique.

Finalement, pour chaque mesure raster KDE créée, il est possible d'associer une seconde mesure sur la bande 2 (Figure 98c). Cette mesure exprime le nombre de points tombant à l'intérieur de chaque pixel (par exemple : le nombre de délits).

(a) Module de création d'un hypercube raster vide (sans mesure): import du fichier xml de définition de l'hypercube

(b) Module de création d'une mesure KDE (densité de points)

(c) Module de création d'une mesure « nombre de points par pixel »

Figure 98. Création d'un hypercube raster KDE

8.3.4. Interface client

La quasi-totalité des opérations SOLAP présentées dans cette thèse sont applicables sur les trois hypercubes London, Seattle et Moon à travers une interface « point & click » programmée en HTML et JavaScript. Cette interface, simplement appelée « Raster Data Cube », est accessible par n'importe quel navigateur via l'adresse web suivante :

<http://nolap01.ulg.ac.be/rastercube>

La suite de cette section fait office de tutoriel pour les lecteurs de cette thèse désirant tester eux-mêmes le prototype. L'adresse ci-dessus conduit à une page permettant à un utilisateur de se connecter au SOLAP. Avant toutes choses, il est nécessaire de se créer un compte utilisateur en cliquant sur le lien « create user ». Une fois connecté, le navigateur affiche l'interface telle que présentée à la Figure 99. L'hypercube London (london_ok.xml) est chargé par défaut, mais il est possible de choisir les deux autres (seattle_ok.xml et moon.xml) en cliquant sur le lien « change data cube » situé au-dessus de la carte.

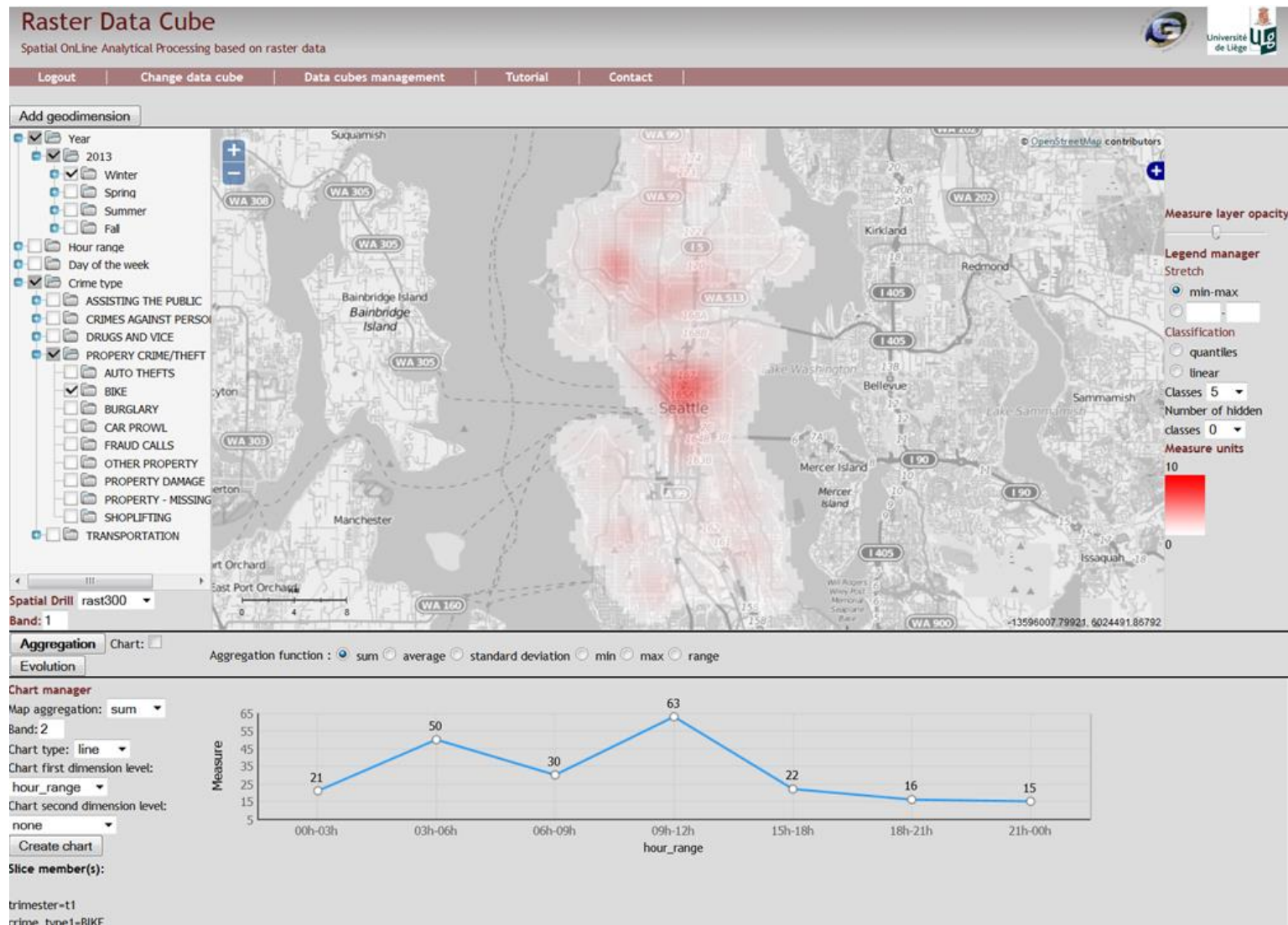


Figure 99. Aperçu général de l'interface client « Raster Data Cube »

8.3.4.1. Interface cartographique

8.3.4.1.1. *Opérations SOLAP : coupes non-spatiales, forages spatiaux et comparaisons cartographiques*

A gauche de la carte, un arbre à dimensions est chargé sur base du fichier de définition du cube (Figure 100a). L'utilisateur a la possibilité d'y sélectionner un ou plusieurs membres de n'importe quel(s) niveau(x) de dimension afin de définir une ou plusieurs coupes non-spatiales (voir section 5.9.1.1). Par exemple, dans la Figure 100a (hypercube Seattle), puisque le membre « Winter » et le membre « Drugs and vice » sont sélectionnés, la carte générée serait le résultat d'une agrégation des types de délits liés à la drogue pour le premier trimestre (hiver).

En dessous de l'arbre (Figure 100b), le menu déroulant « Spatial drill » permet la sélection de la mesure ROLAP correspondant au niveau de forage spatial désiré par l'utilisateur. Pour London et Seattle, ces niveaux portent le nom « rast » suivi de la taille des pixels en mètres. La mesure ROLAP « rast1000 » correspond par exemple à l'hypercube matérialisant le niveau « pixel = 1000 m ». Dans un premier temps, il est conseillé de sélectionner un niveau grossier (rast1000 ou rast300) en l'absence de coupe spatiale, en particulier pour le cube Seattle. En dessous du menu de forage spatial, le choix de la bande correspond à la mesure à agréger. Pour London et Seattle, la bande 1 est la densité de délits (KDE) et la bande 2 est le nombre de délits par pixel. Seule la bande 1 est utilisée pour le cube Moon.

Dans le menu de la Figure 100d, l'utilisateur peut choisir une fonction d'agrégation non-spatiale parmi la somme, la moyenne, l'écart-type, le minimum, le maximum ou l'étendue. Les fonctions « somme » et « moyenne » sont les plus utiles. Les autres fonctions dont l'utilisation est plus délicate feront l'objet de la section 8.4.

Le bouton « Aggregation » de la Figure 100c lance le calcul de la carte en fonction des paramètres choisis par l'utilisateur décrits ci-dessus. Ces paramètres sont envoyés au serveur SOLAP PHP qui construit la requête SQL calculée par l'entrepôt PostGIS. En fonction de la complexité de la requête, la carte est retournée par le serveur après un délai variant entre 1 et 30 seconde(s). Notons qu'en cochant la case « chart », le bouton « Aggregation » génère un graphique en plus de la carte (voir section 8.3.4.2).

Le bouton « Evolution » (Figure 100c), quant à lui, lance une opération SOLAP de comparaison cartographique telle qu'expliquée à la section 5.9.3.1. La comparaison est toujours effectuée entre les deux derniers résultats calculés par l'utilisateur. Par exemple, pour générer une carte montrant l'évolution de la criminalité entre janvier et février, une première carte doit être générée à partir d'une coupe sur le membre « janvier », puis une deuxième carte à partir d'une coupe sur le membre « février ». Le bouton « Evolution » génère ensuite la comparaison cartographique entre ces deux coupes. Remarquons qu'une comparaison ne se limite pas qu'à la dimension temporelle et peut techniquement avoir lieu sur n'importe quelle dimension non-spatiale.

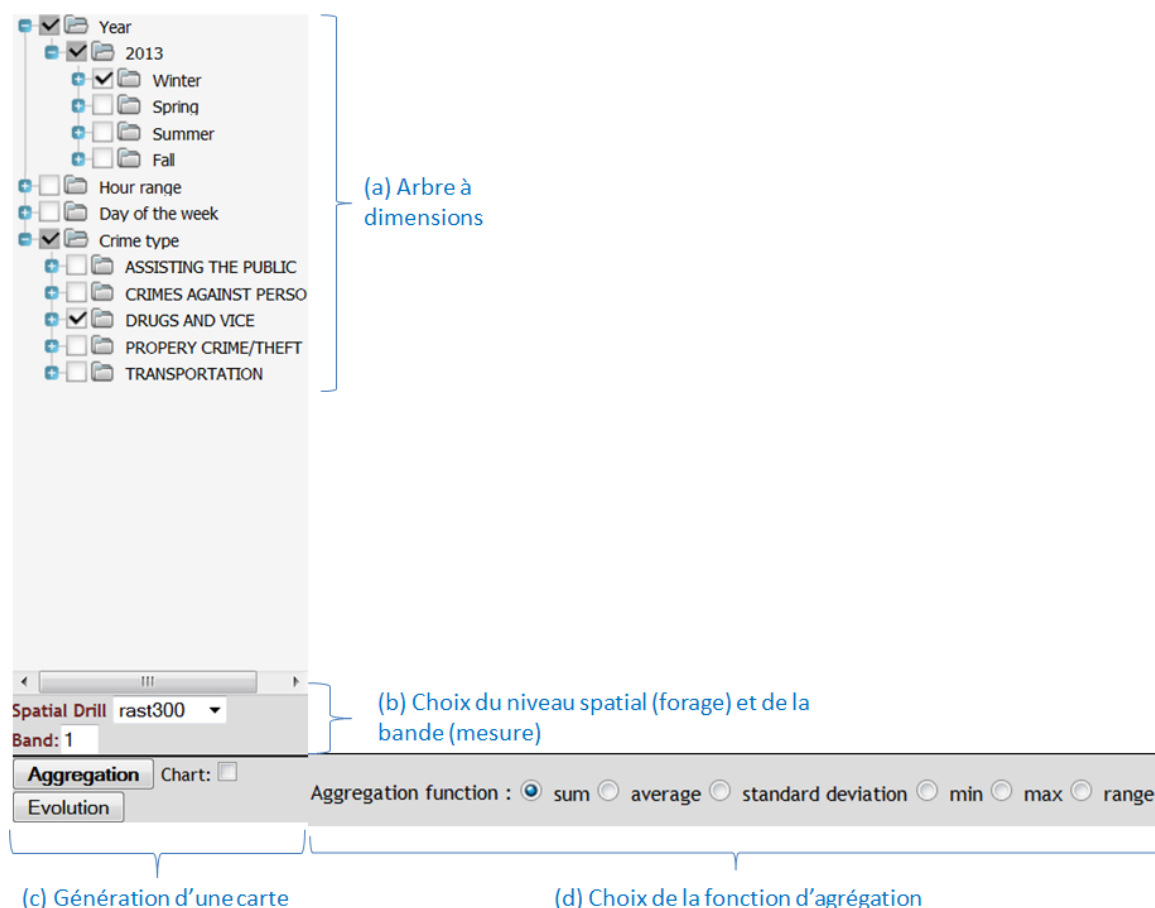


Figure 100. Paramétrage d'une opération SOLAP

8.3.4.1.2. Navigation cartographique et gestion des couches

La carte, gérée par la librairie JavaScript OpenLayers, offre les fonctionnalités de base de navigation telles que « pan » et « zoom » de manière intuitive (clic pour le « pan » et molette de la souris pour le « zoom »). Un gestionnaire de couches est accessible en cliquant sur le « + » du bord supérieur droit de la carte (Figure 101). Le prototype contient les 5 couches suivantes :

- OSM : fond de carte WMS fourni par Open Street Map (dans le cas du cube Moon, il s'agit d'une image géoréférencée de la Lune fournie par la NASA)
- Geodimension : couche des membres spatialisés digitalisés (voir section 8.3.4.1.4)
- Geolabel : couche contenant les toponymes des membres spatialisés digitalisés (voir section 8.3.4.1.4)
- New geomember : membre spatialisé en cours de digitalisation (voir section 8.3.4.1.4)
- Measure : dernière couche raster calculée par le serveur SOLAP que nous appelons simplement « couche SOLAP » (nombre de délits ou densité de délits pour London et Seattle, température pour Moon)

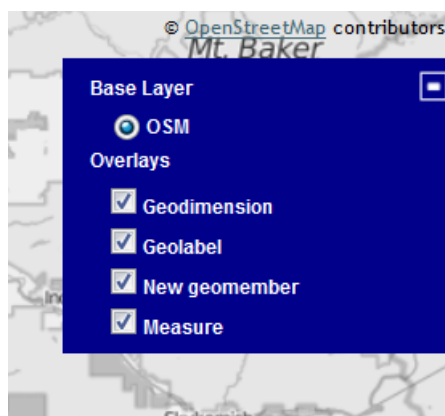


Figure 101. Gestionnaire de couches OpenLayers

Notons que dans l'interface cartographique gérée par OpenLayers, toutes les couches doivent être projetées à la volée dans le SCR du WMS faisant office de fond de carte (« base layer »), à savoir Mercator sphérique, ou Google Mercator (EPSG : 900913), pour le WMS Open Street Map. Pour le WMS de la NASA (cube Moon), nous avons considéré que les données étaient non-projetées dans le SCR WGS84 (EPSG : 4326).

8.3.4.1.3. Symbolisation de la carte

A droite de la carte, se trouve le menu de symbolisation représenté par la Figure 102. Celui-ci permet tout d'abord de régler l'opacité de la couche SOLAP en temps réel (Figure 102a). Plus bas (Figure 102b), l'utilisateur choisit la classification de la couche SOLAP parmi les méthodes suivantes (les couleurs sont imposées par l'interface) :

- Étirement linéaire basé sur les valeurs minimale et maximale du raster de la couche SOLAP
- Étirement linéaire basé sur deux valeurs définies par l'utilisateur
- Classification par quantiles
- Classification linéaire (également basée sur les extrema du raster de la couche SOLAP)

Pour les deux dernières méthodes, l'utilisateur choisit un nombre de classes à calculer (5 par défaut) et un nombre de classes invisibles. Les classes invisibles démarrent toujours de la classe la plus basse. Par exemple, si l'utilisateur choisit 4 classes invisibles pour une classification par quantiles selon 5 classes, seule la meilleure classe (hotspots) est affichée dans la couche SOLAP.

Il est important de signaler que les changements au niveau de la classification ne se font pas en temps réel. Toute modification nécessite un nouveau calcul de l'opération SOLAP (bouton « Aggrégation ») pour être prise en compte dans la couche SOLAP. Après chaque agrégation, la légende de la couche SOLAP apparaît à l'emplacement de la Figure 102c.

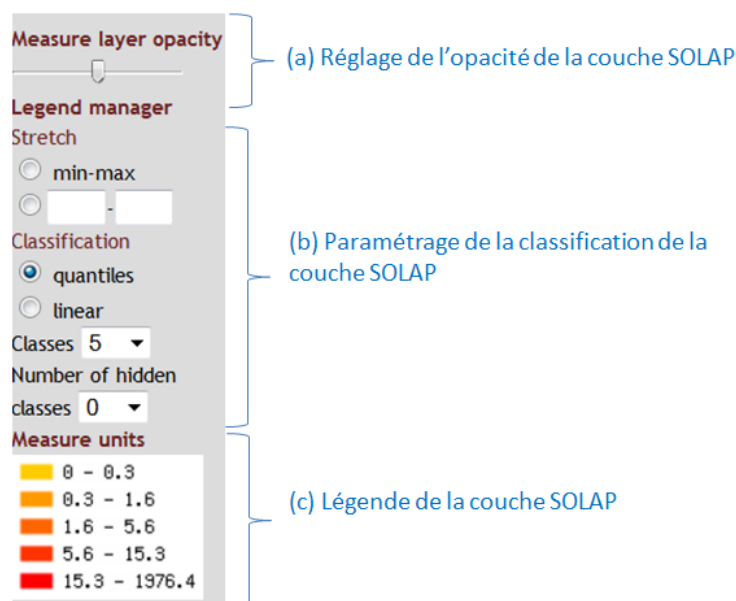


Figure 102. Paramétrage de classification et légende

Notons qu'à chaque requête, tous les paramètres de classification sont transmis au serveur SOLAP qui se charge d'éditer le « map file » du WMS interprété par MapServer.

8.3.4.1.4. Membres spatialisés et opérations SOLAP de coupes spatiales

Au-dessus de la carte, se trouve le gestionnaire des membres spatialisés (ou geodimensions). Celui-ci permet à l'utilisateur de digitaliser des membres spatialisés (ou geomembres) et de les intégrer à la volée dans l'analyse SOLAP. La Figure 103 illustre les étapes de création de deux geomembres sur l'hypercube Seattle. Après avoir pressé le bouton « add geodimension », l'utilisateur digitalise un polygone¹⁸⁰ représentant le Nord de Seattle sur la carte. La fermeture du polygone s'opère par double clic. En cliquant ensuite sur « Record geomember », un pop-up demande à l'utilisateur de nommer le membre ainsi digitalisé avant de l'enregistrer. Il est ensuite transmis au serveur SOLAP qui le stocke dans l'entrepôt de données (table geodimension). La même procédure est appliquée pour enregistrer un geomembre représentant le Sud de Seattle.

¹⁸⁰ Il est possible de digitaliser plusieurs polygones pour modéliser un membre spatialisé. Il est également possible de digitaliser un trou dans un polygone en maintenant la touche « alt » du clavier.

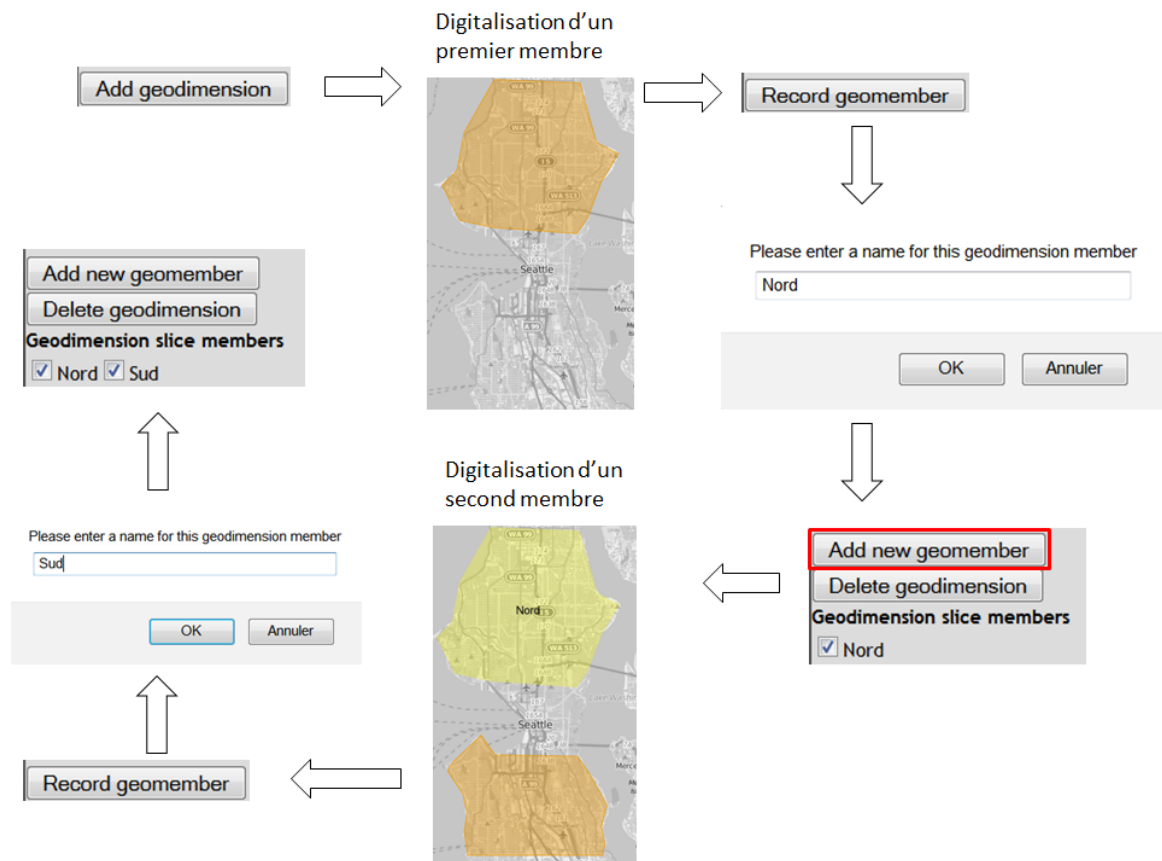


Figure 103. Digitalisation de deux membres spatialisés

A la fin des traitements, les deux geomembres « Nord » et « Sud » apparaissent au-dessus de l'arbre à dimensions et peuvent être cochés, de la même manière que les membres non-spatiaux de l'arbre, pour définir des coupes spatiales. La Figure 104 illustre une coupe spatiale par sélection des deux geomembres après avoir pressé le bouton « Aggregation ».

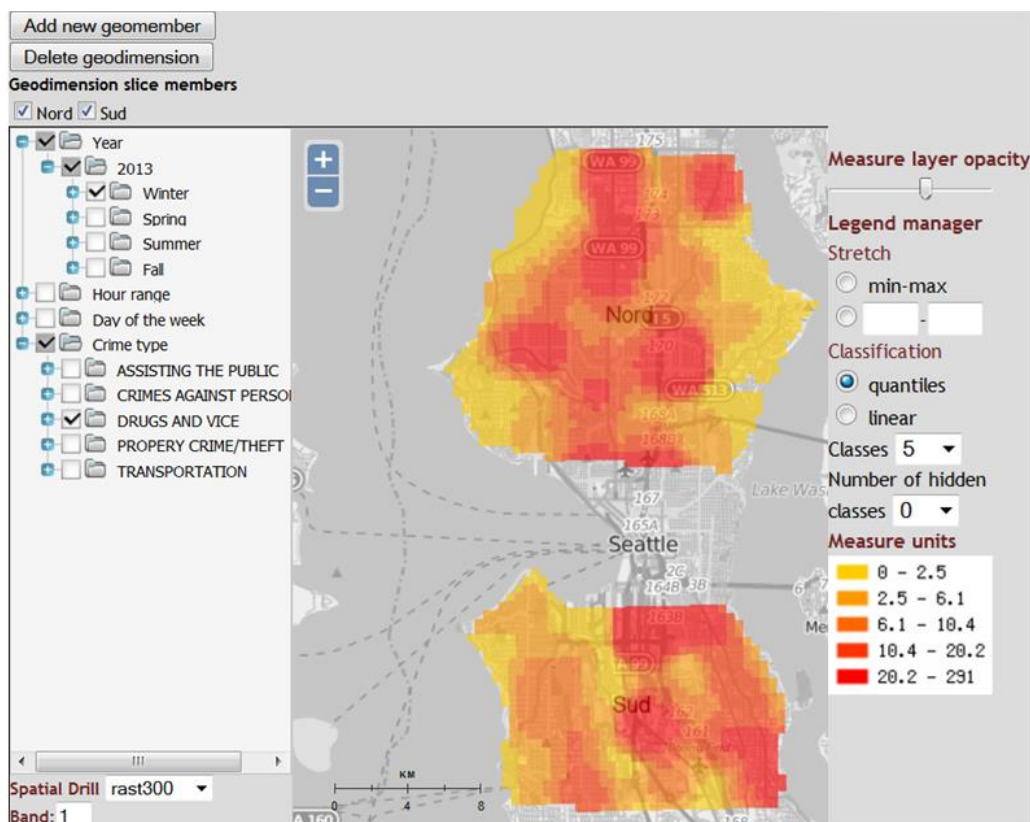


Figure 104. Coupe spatiale selon les deux geomembres digitalisés

8.3.4.2. Interface graphique

En-dessous de l'interface cartographique, l'interface graphique permet l'affichage de dimensions non-spatiales et/ou de membres spatialisés. A l'exception de la symbolisation cartographique, tous les paramètres décrits dans les sections précédentes ont un impact aussi bien sur la carte que sur les graphiques : coupes (non-spatiales et spatiales) dans l'arbre à dimensions, forage spatial (mesure ROLAP), et fonction d'agrégation non-spatiale. Les paramètres propres au graphique sont les suivants (Figure 105) :

- Fonction d'agrégation spatiale (« Map aggregation »)
- Bande contenant la mesure à agréger
- Type de graphique (linéaire, camemberts, etc.)
- Niveau de dimension « intra graphique » affichée (« Chart first dimension level »)
- Niveau de dimension « inter graphiques » affichée (« Chart second dimension level »)

Le bouton « Create chart » lance l'opération SOLAP construisant les graphiques à partir des paramètres choisis par l'utilisateur. Dans l'exemple de la Figure 105, les graphiques affichent les jours de la semaine comme dimension intra graphique. Les paramètres de coupe, de forage spatial et de fonction d'agrégation non-spatiale sont identiques à ceux utilisés dans l'interface cartographique lors de l'opération de la Figure 104. Les données sont donc agrégées pour le premier trimestre et le type de délit « drugs and vice » (coupes non-spatiales), ainsi que pour les deux geomembres « Nord » et « Sud » (coupe spatiale). Puisque la bande 2 est sélectionnée, la mesure traitée est le nombre de

délits, et cette mesure est agrégée selon la somme spatialement (« map aggregation ») et non-spatialement (paramètre de la Figure 100d).

Dans la Figure 105a, la seconde dimension affichée (dimension inter graphiques) est le mois et dans la Figure 105b, il s'agit des membres spatialisés « Nord » et « Sud ». Ces deux opérations SOLAP résultent donc en une comparaison du nombre de délits de type « drugs and vice » par jour de la semaine, respectivement pour chaque mois du premier trimestre et pour chaque geomembre (« Nord » et « Sud »).

Notons que les graphiques sont calculés à partir des raster dont la résolution correspond au niveau de forage spatial choisi dans l'interface cartographique. Dans la Figure 105, des pixels de 300 mètres ont été agrégés alors qu'un résultat similaire aurait pu être obtenu plus rapidement en exploitant le niveau « pixel = 1000 m ». Un niveau de résolution plus grossier engendre cependant une perte de précision sur la définition géométrique des membres spatialisés.

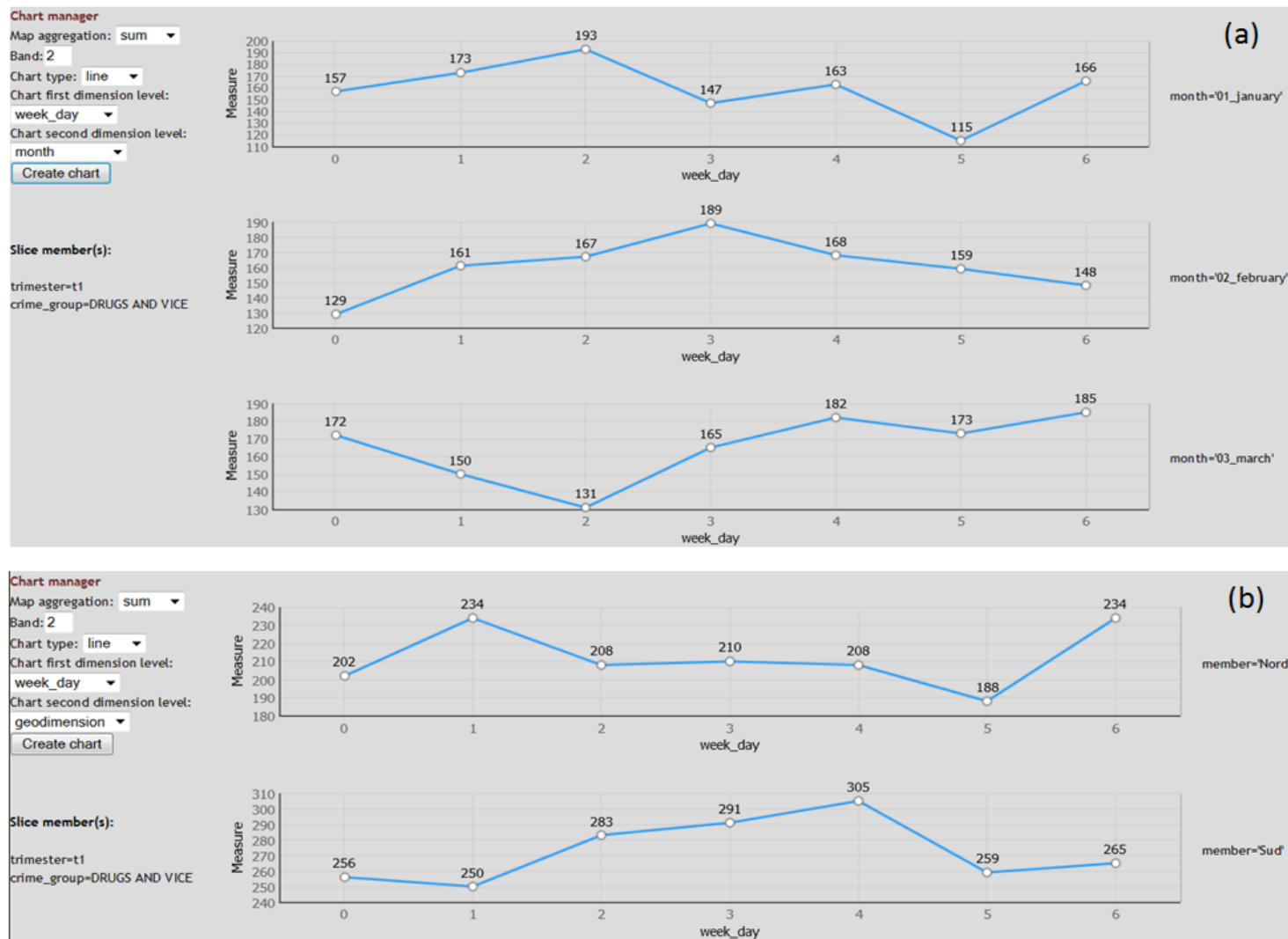


Figure 105. Interface graphique affichant deux dimensions non-spatiales (a) et deux membres spatialisés (b)

8.4. SOLAP raster et agrégations holistiques

Jusqu'à présent, les opérations SOLAP que nous avons illustrées dans cette thèse impliquaient uniquement des fonctions d'agrégation distributives (telles que la somme) ou algébriques (telles que la moyenne). L'emploi de fonctions plus particulières telles que l'étendue ou l'écart-type (fonctions holistiques) peut cependant s'avérer assez intéressant dans le contexte d'un SOLAP raster, et plus particulièrement d'un SOLAP raster continu.

Par exemple, l'agrégation non-spatiale par l'étendue se montre relativement efficace dans l'interface cartographique pour faire ressortir des zones géographiques (hotspots) dans lesquelles la variabilité de la mesure étudiée est plus élevée qu'ailleurs. La Figure 106 illustre une opération de ce type dans l'hypercube London. Un cube raster de la densité de cambriolages (mesure KDE) y est agrégé par l'étendue le long de la dimension temporelle sur l'interface cartographique (Figure 106a). Cette opération fait ressortir très clairement un hotspot sur le quartier de Southall. En effectuant une coupe spatiale ciblée sur ce hotspot et en agrégeant ce sous-cube des cambriolages par la somme sur la mesure « nombre de délits », l'affichage de la dimension temporelle dans l'interface graphique confirme un important pic de cambriolages au mois de juillet pour ce quartier (Figure 106b).

Cet exemple démontre bien l'intérêt des fonctions holistiques dans un SOLAP raster pour détecter et représenter les variations d'une mesure continue de manière cartographique. Cependant, l'emploi de ce genre de fonction nécessite un traitement particulier puisque leurs résultats varient en fonction de la manière dont sont structurées les données (voir section 1.2.5) et leur interprétation n'est possible que lorsqu'elles sont appliquées sur une seule dimension à la fois. Effectivement, l'exemple des cambriolages de Southall est pertinent car la fonction « étendue » a été appliquée pour agréger uniquement la dimension temporelle, la dimension « type de délit » ayant été réduite au seul membre « cambriolages » par la coupe non-spatiale. Sans la coupe, l'étendue aurait produit une carte montrant les variations à la fois sur la dimension temporelle et la dimension « type de délit », dont l'interprétation est ambiguë. Dans ce cas (absence de coupe), une analyse SOLAP de la variation d'une mesure le long d'une dimension n'est donc possible qu'en isolant l'opération élémentaire d'agrégation totale employant la fonction holistique des autres agrégations employant une fonction distributive ou algébrique. Par exemple, pour analyser la variation temporelle de la mesure pour tous les types de délits confondus à travers une requête SOLAP dans l'hypercube London, l'opération élémentaire d'agrégation totale (fonction « étendue ») sur la dimension temporelle devrait être séparée de l'opération élémentaire d'agrégation totale sur la dimension « type de délit » (fonction « somme ») : on agrège d'abord les types de délits par la somme, puis on agrège les mois par l'étendue. Remarquons finalement que l'analyse de la variation d'une mesure ne se limite pas qu'à la dimension temporelle, mais elle peut potentiellement s'appliquer sur n'importe quelle dimension (par exemple : les types de délits).

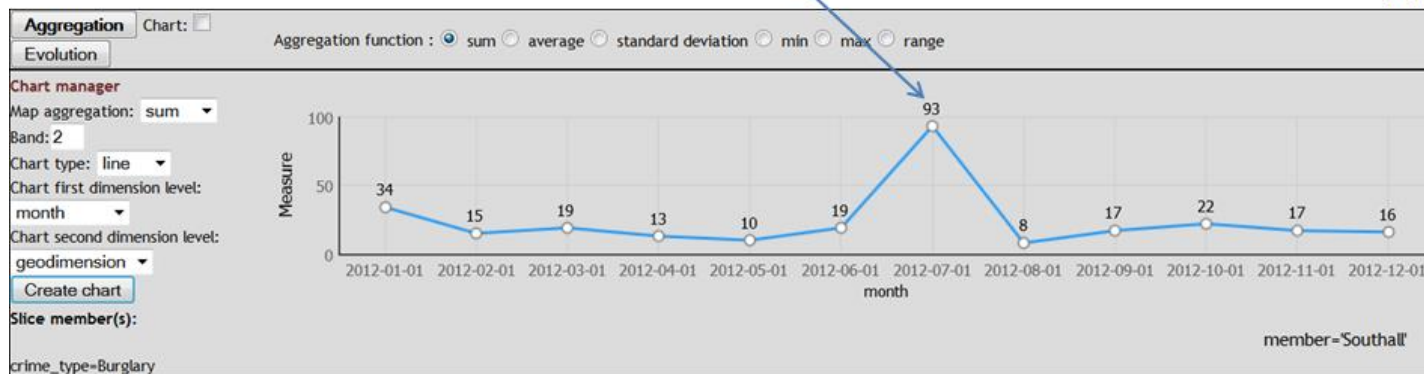
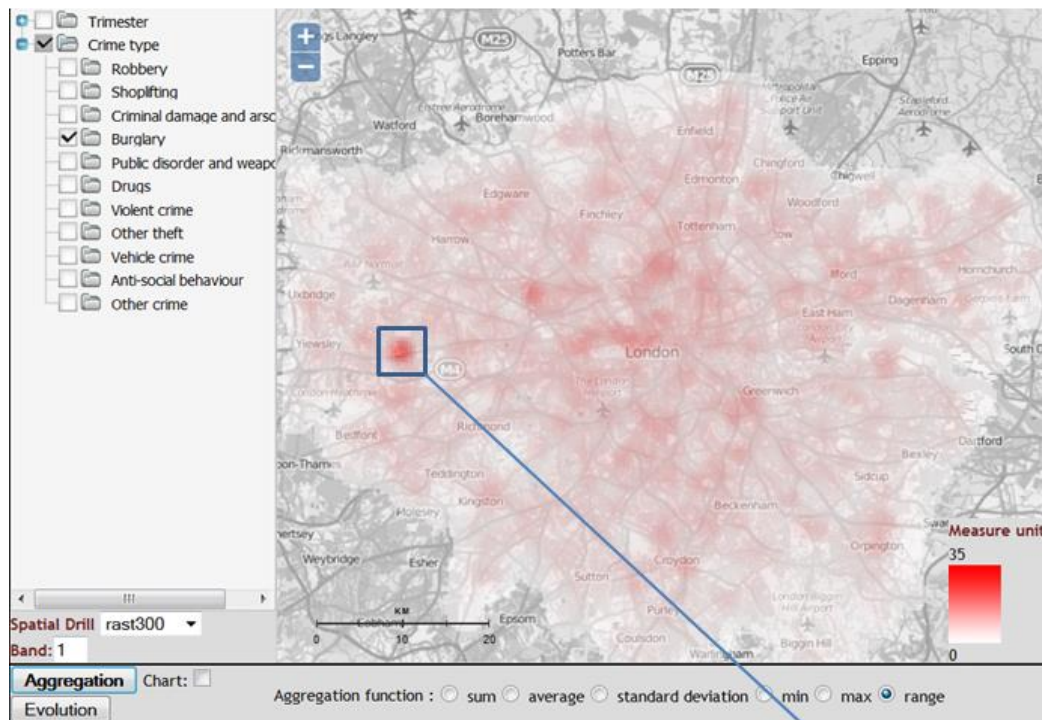


Figure 106. Mise en évidence d'un hotspot avec la fonction « range »

8.5. Étude comparative

Dans cette section, nous présentons et commentons les résultats d'une étude comparative entre trois approches ROLAP spatiales et maillées : l'approche raster, l'approche vectorielle et l'approche SIG. Ces différents types de modélisations OLAP spatiales sont expliqués dans les prochains paragraphes. Cette comparaison va nous permettre de mieux cibler l'utilité du raster par rapport au vecteur dans le SOLAP.

L'approche raster (Figure 107a) est celle étudiée au cours de cette thèse. Elle considère les raster (tableaux où chaque cellule correspond à une maille spatiale) comme des mesures ROLAP et y associe des fonctions d'agrégation spécifiques issues de l'algèbre de carte. Dans Mc Hugh (2008), cette approche est appelée « cube matriciel » (voir section 4.1.1.2). Notons cependant que dans le travail de Mc Hugh, le cube matriciel a uniquement été implémenté à travers le mode vectoriel et non le mode raster (il ne s'agit donc que d'une « simulation » raster).

L'approche vectorielle (Figure 107b) est l'approche SOLAP classique dans laquelle les mailles sont les membres d'une dimension spatiale¹⁸¹. Un niveau d'une dimension spatiale maillée (ou dimension matricielle) est constitué de mailles de taille identique, et le niveau à la résolution la plus fine (niveau le plus détaillé) est relié à la table des faits de manière purement relationnelle comme dans tout schéma en étoile. Ce lien relationnel avec la table de faits implique nécessairement une approche vectorielle dans laquelle une maille est strictement associée à un enregistrement et est modélisée comme un polygone. Dans Mc Hugh (2008), cette approche est appelée « cube objet » (voir section 4.1.1.2). Elle présente deux avantages importants. Premièrement, elle peut être implémentée dans n'importe quel outil SOLAP capable de gérer les géométries vectorielles. Deuxièmement, contrairement à l'approche raster, elle permet l'accès aux données individuelles (enregistrements de la table des faits).

L'approche SIG (Figure 107c), quant à elle, est similaire à une approche SOLAP vectorielle à la différence près qu'il n'y a pas de lien relationnel entre les dimensions spatiales maillées et les enregistrements de la table des faits. Chaque fait est associé à une géométrie vectorielle (par exemple, un point) et le lien avec une dimension spatiale maillée (ou tout autre entité géographique importée à la volée) se calcule par des opérations topologiques (par exemple, l'intersection). En principe, ce genre de calcul à la volée est proscrit dans un SOLAP en raison du très grand nombre d'enregistrements dans la table des faits. La jointure relationnelle de l'approche SOLAP vectorielle est donc généralement préférable pour garantir les temps de réponse courts exigés par l'OLAP (Pendse, 2007). L'approche SIG n'est pas une approche OLAP, mais bien une approche transactionnelle.

8.5.1. Présentation des trois modèles comparés

Dans l'étude comparative de cette section, l'hypercube London a été modélisé selon les trois approches afin de comparer le temps de réponse de ces trois modélisations sur des

¹⁸¹ Nous faisons ici référence aux « dimensions spatiales » telles que considérées dans la littérature SOLAP. Ces dimensions peuvent à la fois présenter des propriétés spatiales et sémantiques.

requêtes SOLAP identiques. Les trois schémas conceptuels (UML étendu) sont illustrés dans la Figure 107. Notons que l’approche SOLAP vectorielle compte deux dimensions spatiales maillées distinctes. L’une comprend les deux niveaux « cell100 » (pixel = 100 m) et « cell300 » (pixel = 300 m), et l’autre comprend uniquement le niveau « cell1000 » (pixel = 1000 m). En effet, il aurait été problématique de lier le niveau « cell300 » au niveau « cell1000 » en raison de la hiérarchie non-strictement engendrée par le chevauchement des cellules d’un niveau à l’autre.

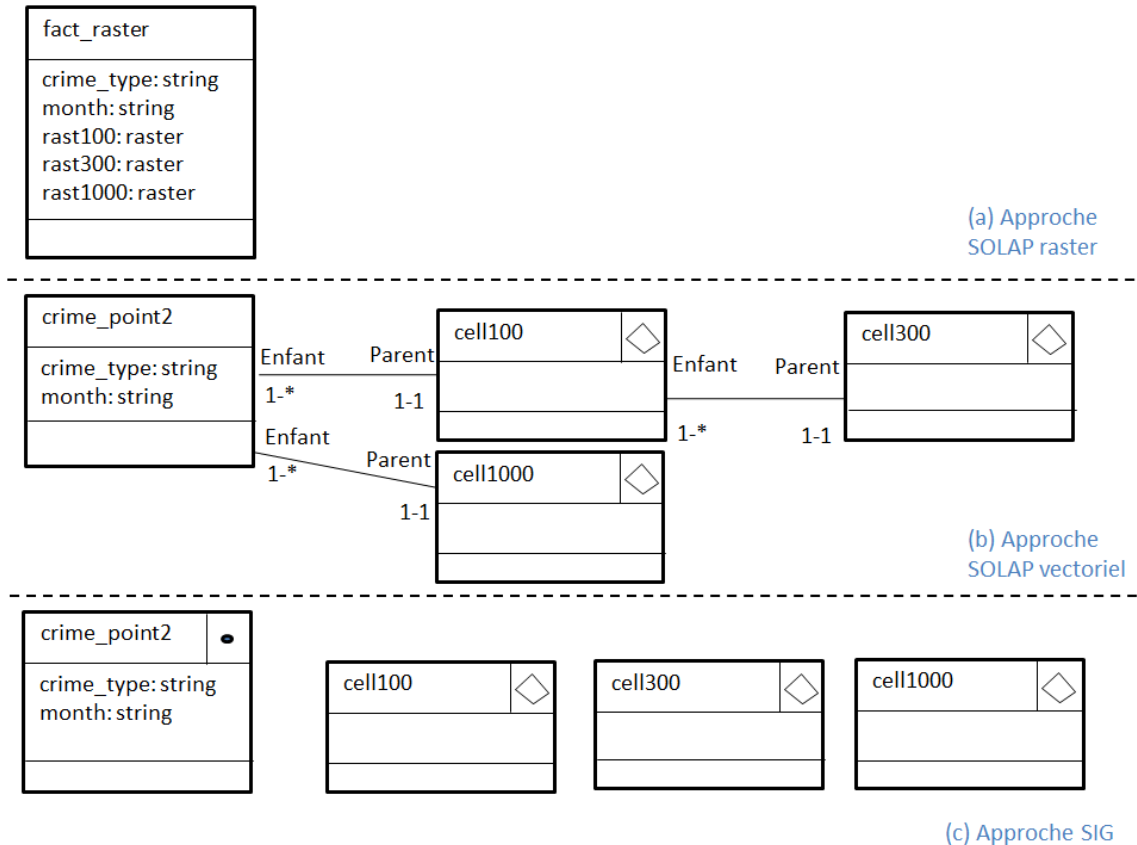


Figure 107. Les schémas en étoile (UML étendu¹⁸²) traduisant les approches SOLAP raster, SOLAP vecteur et SIG pour l’hypercube London

La performance des requêtes SOLAP est avant tout influencée par le nombre de faits détaillés stockés dans l’entrepôt. Les quantités d’enregistrements dans les trois tables des faits respectives sont les suivants :

- Approche SOLAP raster : 132 faits relationnels détaillés
- Approche SOLAP vectorielle et approche SIG : 1 130 646 faits relationnels détaillés

¹⁸² Dans cette modélisation UML, l’entité vectorielle modélisant la définition spatiale d’une classe est représentée par un pictogramme dans le coin supérieur droit.

Le nombre de membres spatiaux (mailles) influence également les requêtes et ce nombre dépend lui-même du niveau de résolution spatial considéré. Dans notre étude comparative, les trois niveaux étudiés sont les suivants :

- Niveau « Pixel = 100 m » : $450 * 585 = 263\,250$ mailles
- Niveau « Pixel = 300 m » : $150 * 195 = 29\,250$ mailles
- Niveau « Pixel = 1000 m » : $45 * 59 = 2655$ mailles

Techniquement, un fait de l'approche raster est un raster si on s'intéresse uniquement à sa composante ROLAP (voir chapitre 6). Si l'on regarde sa composante raster (voir chapitre 5), un fait est défini au niveau du pixel. Par conséquent, le nombre de faits globaux¹⁸³ détaillés pour un niveau donné de l'approche raster est égal au nombre de faits ROLAP (enregistrements de la table des faits) multipliés par le nombre de faits raster (nombre de pixels, ou mailles, d'une mesure ROLAP). Les nombres réels de faits détaillés de l'approche raster pour les différents niveaux spatiaux sont donc les suivants :

- Niveau « Pixel = 100 m » : $263\,250 * 132 = 34\,749\,000$ faits détaillés
- Niveau « Pixel = 300 m » : $29\,250 * 132 = 3\,861\,000$ faits détaillés
- Niveau « Pixel = 1000 m » : $2655 * 132 = 350\,460$ faits détaillés

Finalement, signalons que toutes les géométries traitées dans ces modèles ont été indexées par la méthode GIST (« Generalized Search Tree ») fournie par PostGIS.

8.5.2. Comparaison

27 requêtes ont été exécutées et chronométrées sur les trois modèles stockés dans le même entrepôt de données (géré par PostGIS). Les 15 premières requêtes concernent l'interface cartographique (affichage des dimensions spatiales, ou mailles) et sont répertoriées dans le Tableau 4. Les 12 suivantes concernent l'interface graphique (affichage des dimensions non-spatiales) et sont répertoriées dans le Tableau 5. Les requêtes détaillées en SQL sont présentées dans l'annexe 4. Certaines requêtes utilisent des geomembres digitalisés et stockés dans la table geodimension (voir section 8.3.2.1). Ceux-ci sont représentés dans la Figure 108.

Remarquons que dans le Tableau 5, le niveau de résolution spatiale n'a aucune influence sur les requêtes de l'approche SIG puisqu'elles n'affichent pas de mailles (interface graphique). Les requêtes 22 à 24 ne dépendent pas non plus du niveau spatial pour l'approche SOLAP vectoriel, puisqu'aucun critère spatial n'intervient. En revanche, l'approche raster est toujours influencée par le niveau de résolution, même en l'absence de critère spatial dans la requête.

¹⁸³ Un fait global est une concaténation d'un fait raster et d'un fait ROLAP. Par exemple : la valeur d'un pixel caractérisé par X, Y (fait raster), un mois et un type de délit (fait ROLAP).

Numéro de requête	Dimensions affichées	Coupes	Taille des mailles (mètres)	Temps de réponse (secondes)		
				SOLAP raster	SOLAP vectoriel	SIG
1	X, Y	Aucune	100	15	10	24
2			300	2	5	5
3			1000	0,3	1	3
4	X, Y	Type de délit = cambriolages	100	3	5	36
5			300	0,6	2	24
6			1000	0,1	0,8	17
7	X, Y	Type de délit = cambriolages; Mois = janvier	100	1,5	1,5	5
8			300	0,2	1	4
9			1000	0,1	0,8	3
10	X, Y	4 geomembres	100	11	4	4
11			300	2	2	2
12			1000	0,3	1	1
13	X, Y	4 geomembres ; Type délit = cambriolages	100	2	2	2
14			300	0,2	1	1
15			1000	0,1	0,5	0,5

Tableau 4. Temps de réponse de l'interface cartographique pour les deux approches SOLAP maillées et l'approche transactionnelle (SIG)

Numéro de requête	Dimensions affichées	Coupes	Taille des mailles (mètres)	Temps de réponse (secondes)		
				SOLAP raster	SOLAP vectoriel	SIG
16	Mois	4 geomembres	100	12	2	2
17			300	1,5	1,5	2
18			1000	0,3	0,8	2
19	Mois	4 geomembres ; Type de délit = cambriolages	100	1	1	0,6
20			300	0,3	0,6	0,6
21			1000	0,1	0,5	0,6
22	Type de délit ; mois	Aucune	100	10	20	20
23			300	2	20	20
24			1000	0,1	20	20
25	Type de délit ; mois	4 geomembres	100	10	5	7
26			300	2	5	7
27			1000	0,3	5	7

Tableau 5. Temps de réponse de l'interface graphique/tabulaire pour les deux approches SOLAP maillées et l'approche transactionnelle (SIG)

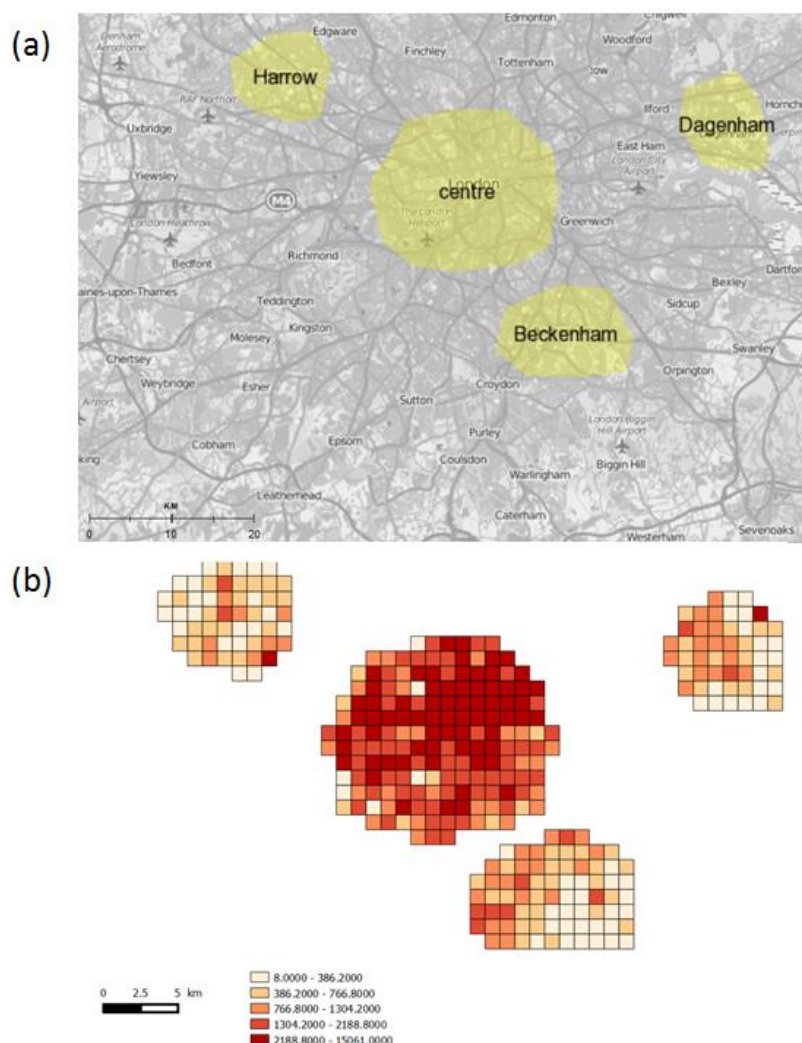


Figure 108. Digitalisation des 4 geomembres (a) et résultat de la requête 12 selon les approches SOLAP vectorielle et SIG (b)

Il est important de souligner que les résultats de cette étude sont purement indicatifs. Pour réellement valider une approche par rapport à une autre, d'autres tests devraient être effectués sur différents jeux de données, différents schémas conceptuels et logiques, et en employant différents outils. Certains choix de modélisation de cette étude peuvent en effet biaiser les résultats. Par exemple, les dimensions non-spatiales « crime_type » et « month » sont dégénérées (incluses directement dans la table des faits) dans les trois approches. Alors que cette dégénération a un impact négligeable dans l'approche raster comptant à peine 132 enregistrements dans la table des faits (un fait = un raster), celle-ci ralentit probablement les temps de réponse de manière significative pour les deux autres approches qui comptent environ 1 millions de faits relationnels stockés (un fait = un délit). D'un autre côté, dans l'approche raster, chaque pixel est codé sur 32 bits¹⁸⁴ alors que les mesures étudiées ici sont des nombres entiers (délits par pixel) pouvant tenir

¹⁸⁴ Ceci est dû au fait que la mesure « nombre de délits » (bande 2) est stockée dans le même raster que la mesure « densité de délits » (bande 1) qui est une valeur réelle.

sur 16, voire 8 bits. Or, comme nous l'avons signalé à la section 2.3.2, le codage des pixels influence considérablement le temps de traitement des raster.

8.5.3. Bilan et discussion

La première constatation de cette comparaison est l'inefficacité du SOLAP raster au niveau spatial le plus fin (« pixel = 100 m »), par rapport aux deux approches vectorielles SOLAP et SIG. En effet, avec PostGIS, la rapidité des opérations locales d'algèbre de carte (agrégations non-spatiales) décroît très vite avec la résolution des raster. En revanche, les niveaux plus grossiers tels que « pixel = 300 m » et « pixel = 1000 m » montrent clairement l'intérêt potentiel de cette approche puisqu'elle affiche les meilleures performances sur toutes les requêtes testées. Cela est principalement dû au fait que dans ce type de modélisation (analyse de données initialement ponctuelles), l'approche raster travaille exclusivement avec des données pré-agrégées sur tous les niveaux spatiaux (pyramides raster). Alors que le nombre d'enregistrements à agréger augmente avec le niveau spatial pour l'approche SOLAP vectorielle¹⁸⁵ et l'approche SIG, le nombre de cellules à agréger décroît pour l'approche SOLAP raster.

En deuxième constatation, on remarque que l'approche SIG affiche des performances pratiquement équivalentes à celles des deux approches SOLAP maillées (raster et vecteur) lorsque les opérations impliquent des coupes spatiales (requêtes 10 à 21 et 25 à 27). PostGIS se montre donc très efficace dans son indexation spatiale puisque les opérations géométriques sont pratiquement aussi rapides que les jointures relationnelles (approche SOLAP vectorielle) ou les découpages raster (approche raster). Précisons néanmoins que l'hypercube de cette comparaison compte « à peine » un million d'enregistrements (il s'agit donc d'un « petit » entrepôt de données) et que les coupes ont été définies sur seulement quatre membres spatialisés. Le léger écart de performance entre filtrage par jointures relationnelles et par opérations spatiales obtenu ici aurait probablement tendance à se creuser davantage en augmentant le nombre de données et de membres dimensionnels. On remarque d'ailleurs que l'approche SIG se retrouve à la traîne pour l'affichage des dimensions spatiales (mailles) dans l'interface cartographique lorsque le niveau spatial est détaillé (requêtes 1 et 4) ou lorsque l'hypercube est peu filtré (requêtes 5, 6, 22, 23 et 24).

La troisième et dernière constatation de ce bilan confirme une de nos hypothèses de recherche. Les agrégations spatiales sont beaucoup plus efficaces avec l'approche raster qu'avec les deux approches vectorielles maillées (SOLAP et SIG). Cela apparaît clairement dans les requêtes 22 à 24 qui affichent les deux dimensions non-spatiales dans l'interface graphique sans coupe. Le calcul de ces 132 faits (11 types de délits * 12 mois) implique une agrégation spatiale des pixels de chaque raster pour l'approche raster. Au niveau le plus détaillé (« pixel = 100 m »), cette opération est deux fois plus rapide avec l'approche raster qu'avec les deux autres approches, alors que le nombre de faits détaillés à agréger est environ 35 fois plus élevé. En effet, pour les approches vectorielles, l'agrégation spatiale de tous les faits de l'entrepôt implique environ 1 million de délits

¹⁸⁵ Notons cependant qu'il n'est pas exclu d'exploiter des cuboïdes matérialisés également dans l'approche SOLAP vectorielle puisqu'il s'agit d'une technique courante dans le domaine de l'OLAP de manière générale.

contre environ 35 millions de pixels pour l'approche raster. Ces agrégations spatiales ont également lieu sur toutes les autres requêtes de l'interface cartographique. Dans ces requêtes, l'écart par rapport aux autres méthodes est moins prononcé en raison de la coupe spatiale vectorielle qui réduit le nombre de données à agréger (avec PostGIS, les coupes spatiales semblent d'ailleurs plus rapides sur du vecteur que sur du raster). Cette efficacité de l'agrégation spatiale sur raster n'est pas sans rappeler les avantages du MOLAP (agrégation en ligne et en colonne dans un tableau) par rapport au ROLAP (agrégation d'une table relationnelle).

En résumé, les expériences menées dans le cadre de cette comparaison montrent que le format raster présente un réel intérêt dans le SOLAP si on le considère comme un magasin de données en aval d'un entrepôt vectoriel (voir Figure 97). En effet, pour l'analyse de données discrètes, il est impossible d'accéder aux données individuelles dans un cube raster alors qu'un cube vectoriel le permet. De plus, les opérations raster sur des niveaux de résolution relativement faibles (avec PostGIS : de l'ordre d'environ 20 000 pixels par raster) présente des performances tout à fait acceptables pour un outil OLAP¹⁸⁶, comme le montrent les requêtes appliquées sur les niveaux « pixel = 300 m » et « pixel = 1000 m » dans l'interface cartographique. Mais la véritable force du raster est sa similarité avec le MOLAP qui le rend beaucoup plus efficace que le vecteur (assimilable à du ROLAP) pour les agrégations spatiales, comme le montrent les requêtes sur l'interface graphique. Bien que non implémentée dans le cadre de cette thèse, on imagine donc qu'en considérant la bande comme une dimension (dimension *B*), les agrégations le long de cette dimension bénéficieraient également des performances du MOLAP, avec un intérêt particulier pour les dimensions continues (voir section 6.8).

Ces résultats nous permettent d'amener une réponse à la question suivante : sous quelles conditions est-il intéressant de produire un magasin de données raster en aval d'un entrepôt vectoriel ? L'étude comparative montre que le raster présente un intérêt uniquement à partir d'un niveau de résolution donné. Comme le MOLAP, le raster souffre du problème de la faible densité (voir section 1.3.5.2), et il est donc intéressant de produire des magasins raster à forte densité par rapport à l'entrepôt auquel ils sont rattachés. Avec les données de l'hypercube London, si pour chaque hypercuboïde raster matérialisant un niveau spatial, on calcule le rapport du nombre total de points de l'entrepôt vectoriel sur le nombre total de pixels, on obtient les densités moyennes de points par pixel suivantes :

- Niveau « Pixel = 30 m¹⁸⁷ » : $1\,130\,646 / 386\,100\,000 = 0,003$
- Densité du niveau « Pixel = 100 m » : $1\,130\,646 / 34\,749\,000 = 0,03$
- Densité du niveau « Pixel = 300 m » : $1\,130\,646 / 3\,861\,000 = 0,29$
- Densité du niveau « Pixel = 1000 m » : $1\,130\,646 / 350\,460 = 3,23$

Théoriquement, les niveaux présentant un réel intérêt devraient présenter un rapport supérieur à 1. Dans notre expérience, nous avons vu que le niveau « pixel = 300 m », affichant un rapport de 0,29, présente pourtant de bonnes performances par rapport aux approches vectorielles. Cette performance accrue est encore une fois imputée aux

¹⁸⁶ Sur les niveaux spatiaux grossiers (1000 m, 300 m), l'hypercube Seattle montre également de bonnes performances malgré le nombre de faits ROLAP détaillés bien plus élevé (environ 11 000 contre 132 pour l'hypercube London).

¹⁸⁷ Ce niveau n'a pas été testé dans l'étude comparative.

particularités de l'indexation raster en tableau, similaire à celle du MOLAP. Avec les outils que nous utilisons, on estime donc qu'un hypercube raster devient rentable lorsque sa densité moyenne de points par pixel présente une valeur de l'ordre de 0,1 ou supérieure.

En transposant ces résultats sur l'hypercube Seattle, on obtient les valeurs suivantes :

- Densité du niveau « Pixel = 30 m » : $806\,663 / (525 * 1194 * 11\,304) = 0,0001$
- Densité du niveau « Pixel = 100 m » : $806\,663 / (158 * 359 * 11\,304) = 0,001$
- Densité du niveau « Pixel = 300 m » : $806\,663 / (53 * 120 * 11\,304) = 0,01$
- Densité du niveau « Pixel = 1000 m » : $806\,663 / (16 * 36 * 11\,304) = 0,12$

Malgré un nombre de faits vectoriels (points) du même ordre de grandeur (806 663 pour Seattle contre 1 130 646 pour Londres), un territoire moins étendu que celui de Londres (qui comprend également sa périphérie), et donc des raster comportant moins de pixels à résolutions spatiales égales, les valeurs de densité de l'hypercube Seattle sont nettement moins favorables que celles de l'hypercube de l'étude comparative. Cela est causé par la différence du nombre de combinaisons possibles de dimensions non-spatiales. L'hypercube Seattle, qui comporte 4 dimensions non-spatiales (type de délit, mois, jour de la semaine et tranche horaire) affiche une table des faits de 11 304 enregistrements (faits ROLAP) contre seulement 132 pour l'hypercube London (qui ne comporte que deux dimensions non-spatiales). Puisque d'après les résultats de l'étude comparative sous PostGIS, un SOLAP raster modélisant un phénomène discret et ponctuel est rentable à partir d'une densité moyenne de l'ordre de 0,1, on peut donc dire que l'hypercube Seattle ne devient rentable qu'à partir du niveau « pixel = 1000 m » (densité de 0,12). Pour obtenir une rentabilité sur des niveaux spatiaux plus détaillés, un nouvel hypercube raster devrait être construit en omettant une ou plusieurs dimensions non-spatiales (par exemple : le jour de la semaine), en supprimant certains membres de dimensions (on pourrait par exemple s'intéresser seulement à quelques types de délits particuliers tels que les vols ou les accidents de la route), ou encore en réduisant le territoire d'analyse (réduction du nombre de membres spatiaux).

8.6. Perspectives

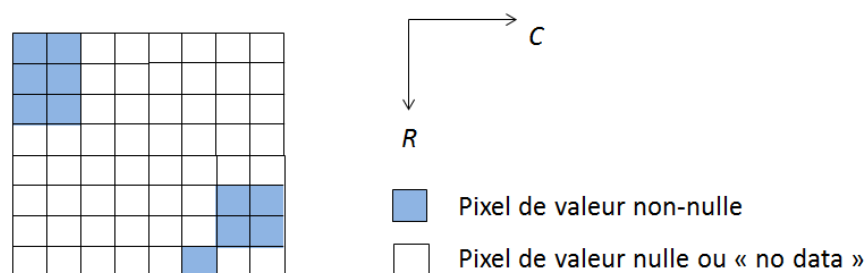
Les trois prochaines sections sont consacrées à trois perspectives qui n'ont pas pu être approfondies dans ce travail : l'hypercube raster tuilé, le NOLAP (« network OLAP ») et le cube raster dimensionnel.

8.6.1. Hypercube raster tuilé

Le tuilage des raster a déjà été abordé dans l'état de l'art de notre question de recherche (section 2.3.2.1), mais pas dans notre modélisation (cube raster et hypercube raster) ni dans notre validation. Cette méthode présente pourtant un certain intérêt dans un hypercube raster puisqu'elle permet l'amélioration de la gestion de la faible densité des dimensions raster. Pour rappel, comme dans les cubes MOLAP, les faits (pixels) d'un cube raster (ou mesure ROLAP dans le cas d'un hypercube) qui ne sont associés à aucune mesure raster occupent tout de même une place dans le cube. En revanche, dans un hypercube raster, les faits ROLAP qui ne sont associés à aucune mesure n'occupent aucune place dans l'entrepôt de données (gestion de la faible densité des dimensions ROLAP uniquement).

La Figure 109a illustre cette mauvaise gestion de la faible densité des dimensions R , C quand un raster est non tuilé. Dans la Figure 109b, le même raster est modélisé par des tuiles composées chacune de 4 rangées et 4 colonnes. On y observe clairement qu'un nombre important de pixels associés à une valeur nulle (ou « no data ») est compris dans une tuile vide. Or, une tuile vide occupe nettement moins d'espace que les pixels vides d'un raster non-tuillé et par conséquent, les faits qui lui sont associés sont traités plus efficacement.

(a) Raster non-tuillé



(b) Raster tuilé

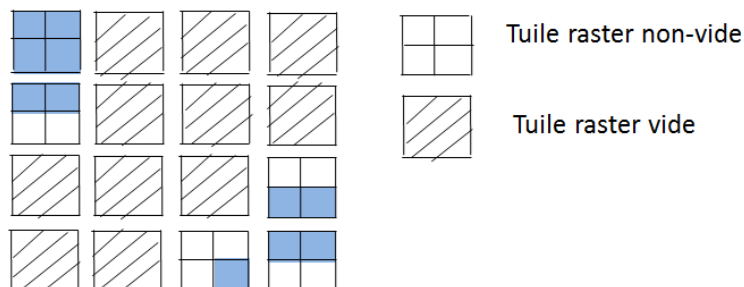


Figure 109. Illustration de la gestion de la faible densité des dimensions raster dans un raster tuilé

Le concept de « tuile vide » est très différent d'un SGBD à l'autre. Avec PostGIS (section 2.4.2), puisque les notions de tuile et de raster sont confondues, une tuile vide n'occuperait aucun enregistrement de la table des faits d'un hypercube raster, de la même manière que les mesures ROLAP associées à un fait ROLAP vide. Une tuile vide PostGIS n'est donc définie nulle part. Même si cette conception peut paraître efficace au premier abord, la nécessité de définir chaque tuile non-vide séparément entraîne une grande redondance des métadonnées raster puisque celles-ci doivent être répétées pour chaque tuile (Figure 22). Dans un hypercube raster, ces métadonnées correspondraient entre autres aux métadonnées spatiales et dimensionnelles décrites aux sections 5.2 et 6.4.

Du côté d'Oracle, nous avons vu à la section 2.4.1 que les métadonnées d'un raster sont clairement séparées de sa grille numérique. Lorsque ce raster est tuilé, chaque tuile occupe un enregistrement de la table « raster data ». Les enregistrements de cette table « raster data » sont reliés à un objet SDO_GEORASTER qui contient les métadonnées de toutes les tuiles sans redondance (Figure 21). Dans un hypercube raster, cet « objet » correspondrait au champ occupé par une mesure ROLAP de la table des faits. Les métadonnées dimensionnelles et spatiales seraient ainsi définies une seule fois pour toutes les tuiles d'une mesure ROLAP. Dans la table « raster data », une tuile vide occupe un enregistrement spécial dont le BLOB (ou grille numérique) est de longueur nulle. Ainsi, une tuile vide est physiquement définie, mais la valeur « nulle » remplace la définition individuelle de tous ses pixels (en l'absence de compression), ce qui rend son traitement plus efficace. Ajoutons finalement que contrairement à PostGIS, Oracle permet également la définition des tuiles sur la dimension *B* (bande). Un cube raster pourrait ainsi être défini comme un ensemble de sous-cubes raster aux dimensions régulières (même nombre de membres pour *R*, *C* et *B*).

Avec sa gestion à part des tuiles raster, Oracle permettrait l'implémentation d'un hypercube raster tuilé en utilisant le même schéma logique en étoile que celui décrit dans notre modélisation ROLAP (voir Figure 71) : chaque niveau spatial matérialisé (niveau de pyramide) est modélisé comme un attribut « mesure ROLAP » différent dans la table des faits¹⁸⁸. L'implémentation d'hypercubes raster dans Oracle n'a pas été réalisée dans ce travail, mais constituerait une perspective très intéressante. En revanche, avec PostGIS, le simple schéma en étoile n'est plus optimisé pour un hypercube tuilé puisque pour une même emprise spatiale (emprise de l'hypercube raster), chaque niveau spatial matérialisé compte un nombre différent de tuiles (plus le niveau est fin, plus les tuiles sont nombreuses). Avec une seule table des faits, un niveau spatial non-détaillé contiendrait donc autant d'enregistrements (nombre de tuiles multiplié par le nombre de faits ROLAP détaillés) que le niveau le plus détaillé, ce qui n'a pas de sens. La solution serait d'employer un schéma en constellation comme celui représenté dans la Figure 110 : chaque niveau spatial matérialisé est stocké dans une table des faits différente.

¹⁸⁸ Il serait même possible de ne définir qu'un seul attribut « mesure ROLAP » pour tous les niveaux spatiaux puisque tous les niveaux de pyramide peuvent être définis dans la table « raster data » gérée par Oracle.

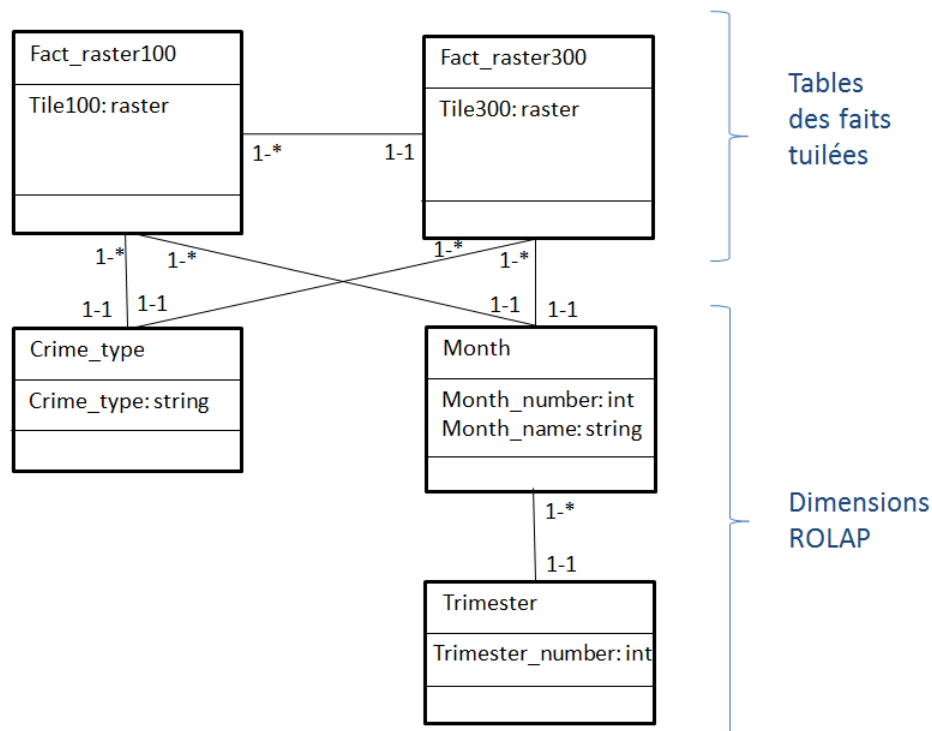


Figure 110. Exemple de schéma en constellation pour un hypercube raster tuilé (UML)

Dans le schéma en constellation de la Figure 110, si l'hypercube raster est tuilé de manière régulière, toutes les tuiles d'une table des faits partagent les mêmes métadonnées dimensionnelles et spatiales, à l'exception des paramètres de translation pour le passage de l'espace raster vers l'espace géographique (nous ne tenons pas compte d'une éventuelle dimension *B* dans cet exemple). Plusieurs essais ont été réalisés dans PostGIS avec ce type de schéma¹⁸⁹. Hélas, les performances atteintes se sont révélées décevantes (raison pour laquelle nous voyons le tuilage comme une perspective). Cependant, lors de ces essais, une seule taille de tuile (nombre de rangées et colonnes) a été testée sur la mesure « rast30 » de l'hypercube London. D'autres tests avec d'autres tailles de tuiles sur de plus petits raster auraient peut-être engendré de meilleurs résultats. Ajoutons que la méthode d'indexation des tuiles proposées par PostGIS n'est peut-être pas la plus pertinente dans ce contexte (méthode « GIST » basée sur l'emprise spatiale vectorielle des tuiles).

Finalement, notons qu'un article paru très récemment (Li *et al*, 2014) propose une méthode inédite pour exploiter efficacement le tuilage d'un cube SOLAP raster à trois dimensions. Dans ce travail, les agrégations raster entre tuiles homologues (dimensions spatiales identiques) sont distribuées vers un « cloud » de plusieurs serveurs qui les calculent en parallèle. Cette distribution des requêtes SOLAP a d'ailleurs également été imaginée dans notre travail (mais de manière différente) et est discutée dans la perspective suivante.

¹⁸⁹ Voir les hypercubes « london_tile30.xml » et « london_tile30_no_empty_tile.xml » dans le prototype « Raster Data Cube ».

8.6.2. « Network OLAP »

Lors de la conception du prototype « Raster Data Cube », une architecture en réseau, intitulée « Network » OLAP (NOLAP), avait été imaginée¹⁹⁰ pour distribuer les requêtes SOLAP vers un « cloud » d'entrepôts de données. En raison des difficultés techniques rencontrées et du manque de temps, cet aspect de notre travail a dû être abandonné. Comme le démontre le travail de Li *et al* (2014), cette perspective demeure néanmoins très intéressante. Dans cette dernière recherche, les requêtes sont divisées en fonction des tuiles d'un cube raster. De notre côté, nous voulions distribuer les requêtes, non pas en fonction des tuiles, mais en fonction des membres d'une dimension ROLAP d'un hypercube raster.

Imaginons par exemple (Figure 111) le cube London où la requête consistant à afficher une carte des hotspots de tous les types de délits confondus pour toute l'année 2012 impliquerait l'agrégation de 120 raster (10 types de délits * 12 mois). Si l'entrepôt de données raster est répliqué dans 10 serveurs, cette requête peut être divisée entre ces 10 serveurs selon la dimension « type de délit ». Le serveur SOLAP maître reçoit la requête client et décompose cette dernière en 10 sous-requêtes agrégeant le temps (12 mois) pour chaque type de délit. Ces 10 agrégations sont effectuées en parallèle par les 10 entrepôts du « cloud » de serveurs. Une fois les 10 résultats retournés au serveur maître, celui-ci les agrège (agrégation de la dimension « type de délit ») en un seul raster qu'il retourne au client. De cette manière, au lieu de prendre le temps d'une agrégation de 120 raster (12 mois * 10 types de délits), la requête prend le temps¹⁹¹ d'une agrégation de 22 raster (12 mois + 10 types de délits).

¹⁹⁰ C'est la raison pour laquelle l'URL de notre prototype contient le terme « nolap01 ». Il existe également deux serveurs supplémentaires « nolap02 » et « nolap03 ».

¹⁹¹ Il faudrait bien sûr également tenir compte du temps de parcours des raster agrégés (résultats fournis par le « cloud ») dans le réseau.

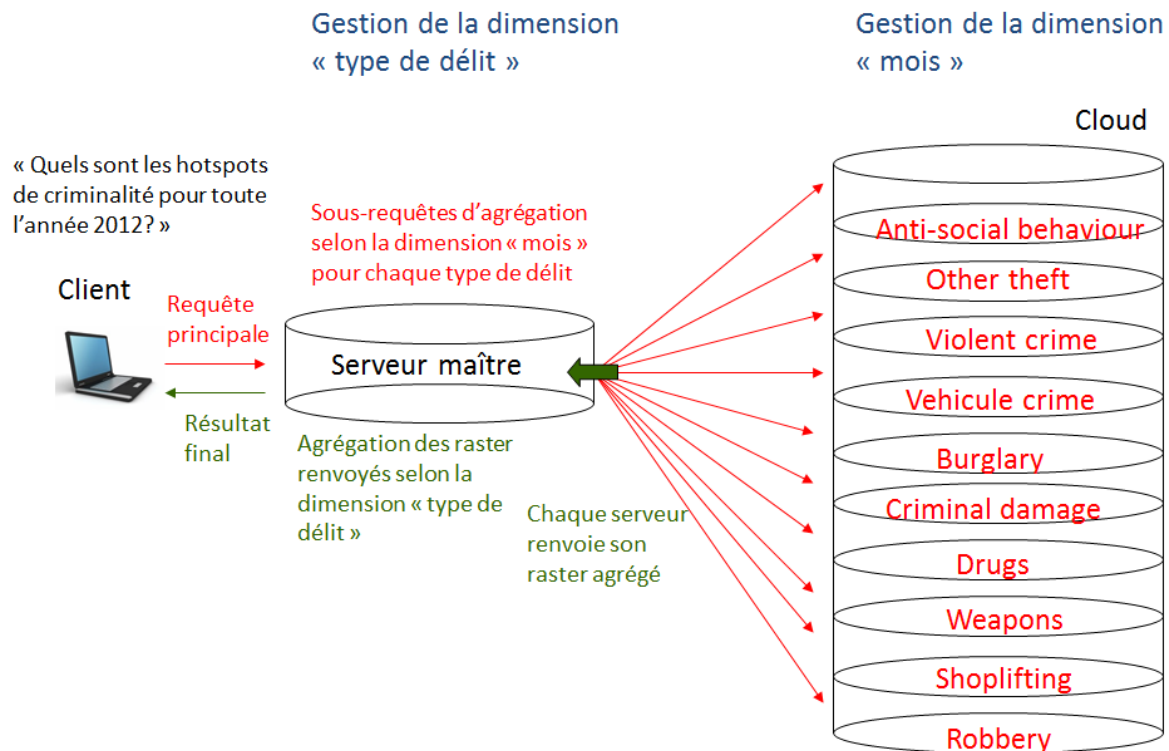


Figure 111. Exemple de requête SOLAP distribuée vers un « cloud » d'entrepôts de données

8.6.3. Cube raster dimensionnel

Jusqu'à présent, l'intégration d'entités géographiques dans un cube raster n'a été envisagée qu'à travers des raster dimensionnels (voir section 5.5.1). Pour rappel, un raster dimensionnel associe des membres spatiaux (pixels) à certaines valeurs sémantiques afin de modéliser des membres spatialisés, ou membres géographiques (par exemple : entités administratives, bâtiments, etc.), intégrables à la volée dans l'analyse de l'utilisateur. Par définition, un raster dimensionnel modélise uniquement les dimensions spatiales R , C . Cependant, il serait également possible d'exploiter des « cube raster dimensionnels » pour importer des membres définis à la fois dans l'espace (dimensions R , C) et dans le temps (dimension B) dans l'analyse de l'utilisateur. Ainsi, les mesures du cube raster de base (ou n'importe quel cuboïde raster matérialisant un niveau spatial) pourraient être agrégées selon des entités géographiques dont la forme et/ou la position est variable dans le temps¹⁹². Avec cette perspective, l'approche raster apporterait une solution originale à la problématique des agrégations SOLAP sur objets mobiles déjà abordée dans la littérature (Kuijpers & Vaisman, 2007 ; Gomez *et al*, 2008).

Prenons un exemple impliquant un phénomène spatialement continu. Dans le domaine de la climatologie, l'intensité de précipitations (mesure continue) pourrait être analysée en fonction d'orages dont la forme et la position sont variables dans le temps (membres géographiques définis selon les dimensions spatiales et la dimension

¹⁹² Un cube raster dimensionnel permettrait ainsi la modélisation de dimensions continues (temps, espace) selon des structures hiérarchiques complexes et dynamiques (voir section 1.2.1).

temporelle). La méthode d'agrégation d'un cube raster (stockant la mesure « intensité de précipitations ») par rapport à un cube raster dimensionnel (stockant la définition spatio-temporelle d'un membre géographique « orage ») correspondrait à une fonction zonale d'algèbre de carte cubique¹⁹³ telle que définie par Mennis *et al* (2005) et illustrée à la Figure 112. Dans un autre exemple impliquant un phénomène spatialement discret, un cube raster dimensionnel modéliserait des entités spatiales administratives aux frontières à géométrie variable, telles que l'Allemagne avant et après la chute du mur de Berlin en 1989. L'application SOLAP forestière étudiée par Miquel *et al* (2002) est également confrontée à des entités géographiques à géométrie variable (voir section 4.1.1.1).

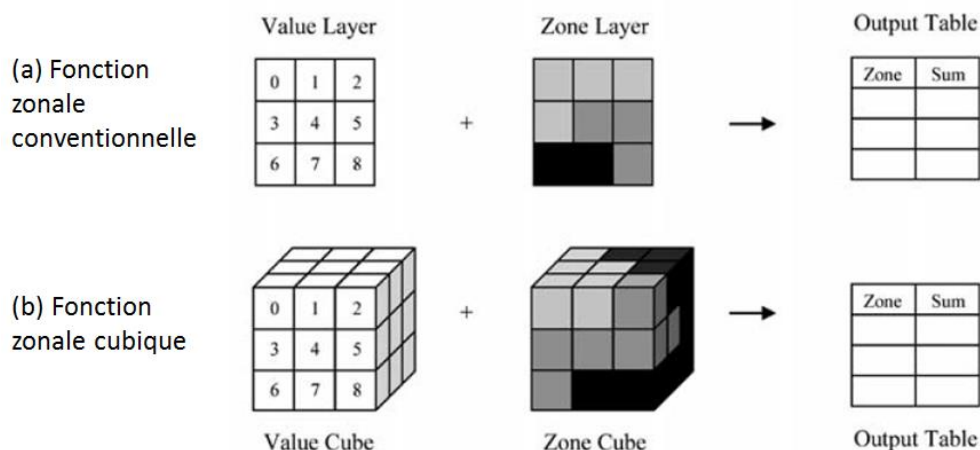


Figure 112. Fonction zonale d'algèbre de carte conventionnelle (a) et cubique (b) (Mennis *et al*, 2005)

Avec cette perspective de cube raster dimensionnel, la notion de cube raster introduite à la section 5.2 mériterait un qualificatif supplémentaire afin d'éviter toute confusion. Le cube raster, qui constitue la base de l'ED et qui est directement dérivé d'un cube MOLAP, pourrait être qualifié de « cube raster de mesures », en opposition au cube raster dimensionnel, ou « cube raster de membres ». Puisque dans un cube raster dimensionnel, les membres sont définis à la fois dans le temps et l'espace, on ne parlerait plus de « membres spatialisés », mais de membres « spatio-temporalisés ».

Ce concept pourrait être poussé encore plus loin en définissant des membres variant dans l'espace, dans le temps, mais aussi dans d'autres dimensions ROLAP. Toujours dans le domaine de la climatologie, imaginons par exemple un orage se déplaçant dans l'espace-temps, mais qui changerait également de catégorie (orage multicellulaire, orage supercellulaire, *etc.*). Ce type de membre serait alors modélisé par un hypercube raster dimensionnel définissant l'espace, le temps et une dimension ROLAP « catégorie ». Dans ce cas, la fonction d'algèbre de carte permettant l'agrégation de la mesure « intensité de précipitation » en fonction de membres définis par un hypercube raster dimensionnel pourrait être qualifiée de fonction zonale d'algèbre de carte hypercubique.

¹⁹³ Notons que Mennis *et al* (2005) définissent également des fonctions focales d'algèbre de carte cubique.

8.7. Conclusion

Dans ce chapitre, notre modélisation SOLAP raster a été validée à travers un prototype exploitant essentiellement les fonctionnalités raster du SGBD PostGIS. Le rôle du serveur SOLAP programmé en aval de l'entrepôt étant essentiellement la construction des requêtes SQL transmises à l'entrepôt de données raster, cet outil rentre dans la catégorie des « SIG-dominants » (voir section 3.5.2). Malgré l'absence de véritable serveur SOLAP (par exemple : GeoMondrian), la richesse des opérations SOLAP atteintes uniquement à travers les fonctionnalités raster de l'entrepôt spatial confirme bien une de nos hypothèses de départ : puisque la structure d'un raster se rapproche de celle d'un cube de données (plus précisément, un cube MOLAP), un outil SOLAP original pourrait être implémenté en exploitant uniquement les fonctionnalités du raster (notamment l'algèbre de carte). Ce type d'outil est cependant très différent des SOLAP vectoriels puisqu'il intègre les dimensions spatiales *X*, *Y* directement dans sa structure multidimensionnelle. Cette particularité permet notamment de gérer les phénomènes spatialement continus, mais aussi d'autoriser l'importation à la volée de membres spatialisés au moment de l'analyse. Ces aspects importants ont pu être validés par les trois hypercubes construits et exploités par le prototype : London, Seattle et Moon.

Remarquons que le prototype présenté dans ce chapitre est assez rudimentaire et pourrait encore être amélioré (il s'agit bien d'un prototype...). Par exemple, les membres spatialisés (geodimensions), qui constituent un atout important de l'approche SOLAP raster (et plus globalement de l'approche SOLAP maillée), ne sont intégrables au moment de l'analyse que par digitalisation. Même si cette digitalisation suffit à prouver la flexibilité de l'outil pour ce type de membre, l'importation de fichiers spatiaux (par exemple : shapefiles) comme couche entière de membres spatialisés constituerait une fonctionnalité particulièrement puissante pour un SOLAP.

Au niveau applicatif, le prototype est capable de gérer l'analyse de hotspots de criminalité à travers la modélisation SOLAP raster KDE (voir chapitre 7). Notre serveur SOLAP construit automatiquement des magasins de données (hypercubes raster en aval d'un ED vectoriel) qui stockent les champs KDE. Ces hypercubes raster permettent à l'utilisateur d'analyser les hotspots de criminalité, selon diverses échelles d'analyse préprogrammées, de manière rapide et conviviale. De plus, l'emploi de fonctions d'agrégaions holistiques telles que l'étendue dans l'agrégaion des champs KDE permet la génération de cartes originales que l'on pourrait qualifier de « cartes de hotspots de variation ». En exploitant les fonctionnalités de l'interface graphique, l'utilisateur a également la possibilité d'approfondir l'analyse des hotspots intéressants révélés par ces cartes, à travers leurs dimensions non-spatiales et leur mesure « nombre de délits ».

A la section 8.5, afin de mieux cerner la place du raster dans le SOLAP, une étude comparative a été dressée entre trois types de modélisations SOLAP maillées sur des données discrètes : SOLAP raster, SOLAP vectoriel et SIG. L'approche SOLAP vectorielle maillée (Mc Hugh, 2008) est avantageuse car elle peut être implémentée facilement dans un SOLAP¹⁹⁴ vectoriel et permet l'accès aux données discrètes individuelles, ce qui est impossible avec l'approche raster développée dans cette thèse.

¹⁹⁴ Pour rappel : tous les outils SOLAP existants sont vectoriels.

L'approche SIG est une approche purement transactionnelle et se montre aussi efficace que le SOLAP uniquement avec de petits jeux de données. Les résultats de l'étude ont montré que l'approche raster affiche de meilleures performances que les deux approches vectorielles (SOLAP et SIG) lorsque la densité moyenne de points de l'ED vectoriel par pixel de l'ED raster est supérieure à une certaine valeur seuil (dans notre cas : de l'ordre de 0,1). Comme nous l'avons montré, la densité moyenne de points par pixel d'un magasin de données raster diminue lorsque le nombre de membres (spatiaux ou non) et/ou le nombre de dimensions diminue(nt). Dans le cas de données discrètes, le SOLAP vectoriel est donc clairement plus adapté aux données plus détaillées, et le SOLAP raster est plus adapté aux données plus agrégées. Cette étude confirme donc la place du SOLAP raster comme magasin de données en aval de l'ED vectoriel pour une analyse de données discrètes (à l'instar du MOLAP en aval du ROLAP).

L'étude comparative nous a également permis de confirmer l'hypothèse que nous avons émise concernant la performance des agrégations spatiales avec l'approche raster (qui se révèlent nettement plus efficaces qu'avec une approche vectorielle). Ce gain de performance du raster par rapport au vecteur s'explique de la même manière que le gain de performance du MOLAP par rapport au ROLAP : dans le contexte de l'OLAP, l'indexation sur des tableaux est plus efficace que l'indexation sur des tables relationnelles. Ce genre de résultat encourage d'ailleurs l'exploitation de la bande des raster comme dimension, approche qui n'a cependant pas été implémentée dans le prototype.

Finalement, trois perspectives de recherche ont été abordées dans la section 8.6. Tout d'abord, nous avons vu que le tuilage d'un hypercube raster pouvait partiellement résoudre le problème de la gestion de la faible densité des dimensions raster. Un schéma en constellation a été proposé comme piste de recherche pour exploiter ces tuiles sous PostGIS. Comme nous l'avons vu à la section 2.3.2.1, le modèle de tuilage dans le SGBD Oracle présente nettement moins de redondance que celui de PostGIS et par conséquent, l'implémentation de notre modèle d'hypercube raster tuilé serait certainement plus performante avec Oracle GeoRaster. Comme démontré par Li *et al* (2014), le tuilage peut aussi être exploité pour distribuer les requêtes SOLAP raster sur plusieurs serveurs afin de réduire considérablement les temps de réponse. En deuxième perspective, nous avons vu que cette méthode d'optimisation serait tout aussi valable en divisant les requêtes selon une dimension ROLAP d'un hypercube raster (tuilé ou non). Enfin, notre troisième et dernière perspective concernait l'exploitation de cubes raster dimensionnels (à ne pas confondre avec un cube raster de mesures) pour intégrer des entités géographiques définies à la fois dans l'espace et dans le temps (intégration à la volée dans l'analyse de l'utilisateur). De cette manière, un hypercube raster apporterait une solution élégante à la problématique du SOLAP en fonction d'entités géographiques mobiles et/ou à géométrie variable. Un exemple a été fourni notamment dans le domaine de la climatologie.

Conclusion générale

Cette thèse s'inscrit dans le domaine de l'informatique décisionnelle, et plus spécifiquement dans celui des entrepôts de données (ED) spatiaux et des systèmes SOLAP (« Spatial On Line Analytical Processing »). Dans un premier temps, notre objectif était d'exploiter ces outils pour subvenir aux besoins des analystes criminels de la Police Fédérale belge. Nous nous sommes cependant rendus compte que les outils SOLAP actuels n'étaient pas adaptés à une méthode très populaire de la cartographie criminelle : la production de cartes de hotspots en estimant la densité par la méthode des noyaux (« Kernel Density Estimation », ou KDE). En effet, KDE implique une vision continue de l'espace afin d'analyser efficacement la distribution spatiale de la criminalité, notamment pour prédire l'emplacement des futurs délits. Or, parmi les outils SOLAP présents sur le marché, aucun n'était capable de gérer cette conception continue de l'espace, ni même d'intégrer des données raster (réputées efficaces pour modéliser un espace continu). Afin d'atteindre notre premier objectif (développement d'un prototype SOLAP pour la police), il devint nécessaire de repenser la conception du SOLAP et des entrepôts de données pour qu'ils intègrent la continuité spatiale et le format raster des données (deuxième objectif de recherche). Par ailleurs, un SOLAP raster continu pourrait également être utilisé dans bien d'autres domaines qui exploitent ce genre de données spatiales : environnement, climatologie, télédétection, *etc.*

Synthèse des différentes parties

La thèse était divisée en trois grandes parties. La **première partie**, intitulée « état de l'art », faisait le tour de la littérature liée à notre objectif SOLAP afin de formuler notre hypothèse de recherche.

Dans le **chapitre 1**, nous introduisons l'OLAP en le replaçant dans son contexte : l'informatique décisionnelle. OLAP est un serveur permettant à un décideur d'analyser manuellement de grands jeux de données hétérogènes à travers des opérations d'agrégation. Pour que cette analyse soit possible, les données provenant de sources diverses sont intégrées via un outil ETL (« Extract, transform, load ») dans un entrepôt de données présentant une structure multidimensionnelle.

Afin de comprendre la structure multidimensionnelle d'un ED, les termes clés de son vocabulaire ont été définis : dimension, membre, hiérarchie, niveau, fait et mesure. Après avoir présenté le schéma en étoile des modèles multidimensionnels au niveau conceptuel, nous avons abordé les différentes manières de traduire ce schéma au niveau logique pour un ED relationnel : schéma en étoile, en flocon, en constellation et en galaxie. La section sur les entrepôts de données et la structure multidimensionnelle a ainsi pu être clôturée par la définition de l'instance d'un schéma multidimensionnel : le cube de données. Celui-ci peut être décomposé en cuboïdes de données pré-agrégées. En effet, nous avons vu que l'approche OLAP (à ne pas confondre avec le serveur OLAP), contrairement à l'approche OLTP, autorisait la redondance de l'information et la

dénormalisation des tables afin d'accélérer le temps de réponse du système, quitte à consommer plus d'espace.

Après avoir introduit les entrepôts de données et la structure multidimensionnelle, nous avons pu définir les caractéristiques du serveur OLAP : opérations de manipulation de cubes de données (forages et coupes), types d'interfaces utilisateur (tables de pivot et graphiques), langage MDX de référence et les trois grands types d'architectures OLAP (MOLAP, ROLAP et HOLAP). Alors que le ROLAP (« Relational OLAP ») utilise un entrepôt de données relationnel, le MOLAP (« Multidimensional OLAP ») exploite les données directement stockées sous forme de tableaux multidimensionnels (indexation des membres dimensionnels par les coordonnées des cellules) : on parle alors d'entrepôt de données (ou de cube de données) physiquement multidimensionnel. Le ROLAP permet le stockage de grands volumes de données détaillés dans des cubes de faible densité, mais le MOLAP délivre des réponses plus rapidement. HOLAP (« Hybrid OLAP ») est une architecture combinant à la fois les caractéristiques du ROLAP (pour les données plus détaillées) et du MOLAP (pour les données plus agrégées).

Le **chapitre 2** était centré sur les systèmes d'information géographique (SIG) car l'OLAP spatial (SOLAP) est une association des concepts OLAP et SIG dans un même système.

Au cours de la première partie de ce chapitre, nous avons défini les concepts de base du SIG en mettant l'accent sur deux dualités : d'un côté les données vectorielles et les données raster (dont la structure physique présente beaucoup de similarités avec un cube de données MOLAP), de l'autre côté les phénomènes spatialement continus et spatialement discrets. Sur base des caractéristiques respectives des modèles vecteur et raster, un bilan a été dressé sur leur application en fonction du caractère discret ou continu d'un phénomène géographique. En particulier, les phénomènes continus semblent très adaptés au raster lorsqu'ils sont modélisés comme des champs réguliers en raison de la gestion implicite de la géométrie et de la topologie. A l'inverse, pour les champs irréguliers, le mode vecteur semble plus adapté en raison de sa souplesse au niveau géométrique. Nous avons ensuite abordé la notion de couverture spatiale telle que définie par l'ISO et l'OGC car elle constitue une norme solide pour la modélisation conceptuelle des phénomènes géographiques discrets et continus, indépendamment du type de modélisation (vecteur ou raster).

La deuxième partie du chapitre était centrée sur la modélisation raster qui constitue la base de notre hypothèse de recherche en raison de toutes les similitudes qu'elle comporte avec les cubes MOLAP, ainsi que son aptitude à modéliser efficacement la continuité spatiale. Quelques caractéristiques importantes ont ainsi été définies : l'algèbre de carte, les tuiles et les pyramides. Pour terminer, deux SGBD spatiaux capables de stocker et manipuler des données raster ont été présentés et comparés : Oracle et PostGIS. Cette comparaison a révélé deux modèles physiques aux philosophies sensiblement différentes. D'un côté, Oracle optimise le stockage des tuiles et leur traitement à travers une architecture relativement complexe. De l'autre côté, PostGIS propose une forte interopérabilité entre données raster et vecteur à travers une architecture plus simple et plus souple, mais aussi plus redondante.

Dans le **chapitre 3** nous présentons la technologie SOLAP (association de l'OLAP et du SIG). Ce type d'outil permet à un décideur d'analyser de grands jeux de données agrégés selon plusieurs niveaux de granularité en fonction d'un modèle multidimensionnel, avec une attention particulière portée sur la composante spatiale. Cette analyse s'appuie sur des cartes, des graphiques et des tableaux dans une interface simple et interactive. Afin d'enrichir l'OLAP avec les techniques du SIG, la structure multidimensionnelle des entrepôts de données doit intégrer l'information spatiale (géométrique) dans ses dimensions et/ou ses mesures. Ainsi, les opérations OLAP, redéfinies en opérations SOLAP (forages et coupes), peuvent être appliquées directement sur la carte et sont combinables avec les opérations d'analyse spatiale typiques du domaine du SIG.

La dernière section du chapitre portait sur la classification des différents outils SOLAP actuellement sur le marché (propriétaires et « open source »). Cette distinction est basée sur l'importance relative des composantes OLAP et SIG dans les SOLAP : OLAP dominant, SIG dominant et OLAP-SIG intégré. La littérature montre bien que la catégorie OLAP-SIG intégré représente un idéal vers lequel la technologie SOLAP (qui est encore très jeune puisqu'elle a moins de 20 ans) devrait se diriger afin de permettre à l'utilisateur des analyses toujours plus souples et plus riches au niveau spatial comme au niveau décisionnel. Alors que l'information raster constitue une part très importante du SIG, celle-ci est pratiquement absente du domaine SOLAP qui devrait pourtant être capable d'intégrer tous les formats de données spatiales dans son analyse. De plus, comme nous l'avons déjà signalé dans notre objectif de recherche, l'information spatialement continue qui représente aussi une part importante du SIG (à un niveau plus conceptuel) n'est pas gérée par les SOLAP actuels.

En clôture de cette première partie de thèse, le **chapitre 4** expliquait notre hypothèse de recherche. D'après la description des outils SOLAP existants fournie dans le chapitre 3, nous avons décelé trois caractéristiques spatiales importantes qui leur font défaut : l'intégration du mode raster, la possibilité de fournir une analyse continue de l'espace et la gestion de la spatialité directement dans un cube MOLAP.

En analysant la littérature scientifique portant sur la première problématique (l'intégration du raster), nous avons remarqué que les quelques travaux traitant du SOLAP maillé ouvraient de solides pistes de modélisation. Deux avantages théoriques importants de ce type de modèle sont la gestion du spatialement continu et la possibilité d'appliquer des opérations spatiales à la volée dans l'analyse de l'utilisateur. Cependant, aucun de ces modèles n'a été implémenté en conservant la structure physique du raster (tableau multidimensionnel) qui est pourtant un élément clé pour l'optimisation des requêtes spatiales appliquées sur ce type de données. Nous avons également constaté que certains travaux portant sur les bases de données raster proposaient des méthodes intéressantes pour optimiser les agrégations sur les données raster.

Du côté de l'analyse spatialement continue intégrée au SOLAP, l'état de l'art était plus conséquent et son contenu plus varié. En effet, depuis une dizaine d'années, des modèles sont proposés à différents niveaux d'abstraction. Certains sont indépendants du type de modélisation de l'information spatiale (vecteur ou raster) alors que d'autres sont valables uniquement en mode vecteur. Le raster, quant à lui, apparaît pratiquement anecdotique dans le sens où il n'a jamais réellement fait l'objet d'une étude exploitant son potentiel physique dans un SOLAP continu. Pourtant, l'algèbre de carte, qui est pensée à la

base pour ce type de données, est reprise comme méthode d'agrégation des champs dans la majorité des modèles présentés.

Lors du premier chapitre, une comparaison des trois grands types d'architectures OLAP avait été dressée (ROLAP, MOLAP, HOLAP). Nous y avons constaté que l'architecture MOLAP offrait d'importants avantages pour l'optimisation des requêtes OLAP. Dans le chapitre 2, une description de la structure physique du raster nous a clairement montré qu'elle s'apparente à un cube MOLAP. La discussion qui a suivi, portant sur les avantages respectifs du mode vecteur et raster, a démontré que le format raster est bien adapté à l'information spatialement continue.

Partant de ces différents constats, notre hypothèse fut la suivante : puisque le raster est largement utilisé dans les SIG pour modéliser les champs continus, et que sa structure physique s'apparente à un cube de données MOLAP, un outil SOLAP considérant le raster comme un cube de données devrait être capable de gérer une analyse multidimensionnelle de l'information spatialement continue, tout en optimisant les agrégations spatiales. Ce genre de solution, qui devrait tirer profit des fonctions raster implémentées dans les outils SIG, permettrait d'intégrer des membres spatiaux à la volée dans l'analyse de l'utilisateur. La composante spatiale des outils SOLAP serait ainsi enrichie afin de renforcer l'intégration de l'OLAP et du SIG.

La **deuxième partie** de la thèse, la plus importante, décrivait le modèle SOLAP raster théorique développé dans ce travail. Il s'agissait naturellement de la partie la plus originale.

Dans le **chapitre 5**, nous décrivions toutes les caractéristiques d'un modèle SOLAP tridimensionnel exploitant le format raster bien connu du domaine du SIG. Pour établir ce modèle, nous sommes partis de la définition d'un cube MOLAP comportant trois dimensions : *R* (rangée), *C* (colonne) et *B* (bande). En associant des métadonnées spatiales aux dimensions *R* et *C* (espace raster), nous avons démontré que celles-ci pouvaient modéliser des dimensions spatiales planimétriques *X* et *Y* implicitement (de la même manière que dans la modélisation raster propre au SIG). Nous avons simplement appelé ce type de cube « cube raster ». Les cellules de ce cube (pixels), sont considérées comme des faits spatiaux auxquels est associée une mesure, et un membre spatial est à la fois défini par la dimension *R* et la dimension *C*. En nous inspirant des pyramides raster, nous avons démontré qu'une hiérarchie implicite pouvait être définie sur ces dimensions pour modéliser différents niveaux de résolutions spatiales. Afin d'optimiser les opérations SOLAP de forage sur cette hiérarchie (forage spatial), les niveaux sont stockés dans des cuboïdes raster.

Après avoir discuté des caractéristiques des dimensions spatiales *R* et *C*, nous avons défini les caractéristiques de la dimension *B*, habituellement associée à la bande spectrale dans la modélisation raster. Dans notre modèle, celle-ci peut être associée à n'importe quelle autre dimension comme dans tout modèle OLAP. Nous avons cependant souligné le fait qu'il était intéressant de l'associer elle aussi à des métadonnées lui permettant de modéliser implicitement une dimension continue telle que le temps ou la hauteur. Notons que si notre travail s'était limité à un modèle tridimensionnel, la dimension *B* devrait nécessairement modéliser le temps puisqu'un entrepôt de données est avant tout un outil d'archivage (les deux autres dimensions étant obligatoirement des dimensions spatiales).

Puisque la définition des dimensions spatiales X , Y est implicite grâce aux métadonnées du cube raster, n'importe quel raster externe partageant les mêmes métadonnées que le cube peut être mis en relation avec ce dernier pour intégrer une couche de membres spatialisés dans l'analyse de l'utilisateur. Ce que nous appelons « membre spatialisé » est l'équivalent d'un membre d'une dimension spatiale dans la littérature SOLAP au sens large (par exemple : des communes, des routes, *etc.*). Dans notre modèle, il aurait été ambigu d'utiliser le terme « dimension spatiale » puisqu'il fait déjà référence à X et Y . De plus, les membres spatialisés (ou membres géographiques) n'étant pas définis dans le schéma conceptuel en étoile (intégration à la volée), ils ne sont rattachés à aucune dimension prédéfinie comme c'est le cas dans un SOLAP vectoriel. Un raster modélisant ainsi une couche de membres spatialisés est appelé « raster dimensionnel », et lorsque ce raster partage les mêmes métadonnées que le cube, nous l'appelons « raster dimensionnel intégré ». Les opérations nécessaires à cette intégration à la volée dans l'analyse de l'utilisateur ont été décrites de manière précise.

Après avoir illustré notre cube raster par un exemple concret impliquant des données de criminalité de la ville de Londres, nous avons décrit une chaîne d'opérations élémentaires (agrégaions et filtrages) permettant le calcul des mesures des faits affichés dans une interface cartographique ou graphique (la table de pivot n'a pas de raison d'être dans ce modèle puisqu'une cellule détaillée est un pixel raster nécessitant un positionnement sur une carte pour être interprété par l'utilisateur). En particulier, les fonctions d'agrégation sont directement inspirées de l'algèbre de carte et les filtrages exploitent des découpages de tableaux bien connus du domaine raster. Dans les interfaces, l'utilisateur a la possibilité de visualiser tous les types de dimensions (spatiales et non-spatiales) ainsi que les membres spatialisés, et d'exécuter les opérations SOLAP classiques sur un cube raster : coupes, forages, mais aussi comparaisons. Toutes ces opérations présentent une variante spatiale et non-spatiale. Elles ont été décrites au moyen de la chaîne d'opérations élémentaires et ont été illustrées dans le cube raster londonien présenté précédemment. La dernière section de ce chapitre était consacrée à l'indexation physique d'un cube raster qui présente exactement les mêmes caractéristiques que les données raster (entrelacement).

Ce modèle raster exploite la spatialité des faits directement dans la structure multidimensionnelle d'un cube de données à travers le format raster et valide donc en partie notre hypothèse de recherche (considération du raster comme un cube MOLAP). De plus, sa capacité à intégrer n'importe quelle couche de membres spatialisés à la volée lui donne un avantage par rapport à la plupart des modèles SOLAP vectoriels. Le nombre de dimensions non-spatiales du modèle raster multidimensionnel est cependant limité à une seule (nous n'avons pas connaissance de raster à plus d'une dimension « bande » et les cubes MOLAP sont eux-mêmes très limités par le nombre de dimensions).

Le **chapitre 6** introduisait une extension du modèle raster tridimensionnel présenté dans le chapitre 5 afin d'y intégrer un nombre indéfini de dimensions non-spatiales. Ce modèle fut appelé « modèle raster relationnel ». L'idée est de considérer un cube raster comme une mesure d'un modèle ROLAP classique (mesure ROLAP). Les trois dimensions du cube raster (R , C , B) sont ainsi enrichies de dimensions supplémentaires gérées par le modèle ROLAP : les dimensions ROLAP. L'instance d'un tel modèle est donc un hypercube raster. Puisqu'un hypercube raster est composé à la fois de dimensions ROLAP

et de dimensions raster (assimilables à du MOLAP), chaque élément de sa structure multidimensionnelle est défini à la fois dans le modèle raster et dans le modèle relationnel : hiérarchies raster et hiérarchies ROLAP, faits raster et faits ROLAP, mesures raster et mesures ROLAP, mais aussi cuboïdes raster et cuboïdes ROLAP. Puisque les mesures ROLAP sont des cubes raster, celles-ci sont agrégées par une opération de type « drill across » consistant à agréger entre elles les mesures de plusieurs cubes de données partageant les mêmes dimensions raster et la même instance (algèbre de carte cubique). Pour que ces cubes partagent les mêmes dimensions spatiales X , Y et les mêmes instances spatiales x , y , ils doivent nécessairement être définis par les mêmes métadonnées spatiales (SCR , h_R , h_C , t_R , t_C) et dimensionnelles (N_R , N_C). Afin de ne pas alourdir les opérations SOLAP par des transformations raster à la volée, ces métadonnées sont définies comme des contraintes sur les mesures ROLAP dans une table spécifique : la table de métadonnées. Une chaîne d'opérations élémentaires (filtrages et agrégations) permettant le calcul des mesures affichées par l'interface SOLAP intègre les opérations spécifiques aux dimensions ROLAP et les opérations spécifiques aux dimensions raster (appliquées sur un hypercube, et non plus sur un simple cube). Les interfaces et opérations SOLAP dans le modèle relationnel sont sensiblement les mêmes que dans le modèle raster, mis à part que les graphiques sont capables d'afficher plusieurs dimensions non-spatiales dans le modèle relationnel (le modèle purement raster ne pouvait pas manipuler plus d'une dimension non-spatiale). Tout comme le modèle purement raster, le modèle ROLAP est capable d'intégrer des membres spatialisés à la volée.

En résumé, grâce cette extension relationnelle, notre modèle SOLAP raster peut gérer plusieurs dimensions non-spatiales en plus des deux dimensions spatiales X et Y . De plus, il est facile de l'implémenter dans un SGBD intégrant les données raster (par exemple : PostGIS ou Oracle) puisque les opérations élémentaires d'agrégation et filtrage raster correspondent respectivement à de l'algèbre de carte et à des découpages de grilles numériques géoréférencées bien connus du domaine raster.

Le **chapitre 7** étudiait l'intégration de l'information spatialement continue dans le modèle d'hypercube raster développé dans le cadre de cette thèse (premier objectif de recherche). Comme nous l'avons vu dans le chapitre 2, la structure maillée des données raster permet une modélisation simple et efficace des phénomènes spatialement continus (on parle alors de champs raster). Puisque toutes les cellules d'un hypercube raster correspondent en réalité aux pixels d'un raster, le modèle développé lors des précédents chapitres était déjà bien adapté à ce type de phénomène. Un hypercube raster continu modélise donc une couverture spatiale continue où la mesure d'un fait détaillé est une valeur de champ définie par les membres de dimensions (spatiales et non-spatiales) associés au fait. Les calculs des faits non-détaillés sont effectués uniquement par agrégation des champs raster intégrés à l'entrepôt. Contrairement aux modèles SOLAP continus exploitant le format vectoriel, tous les calculs d'interpolation des champs sont réservés uniquement à la phase d'intégration des données et non à la phase d'analyse.

La suite du chapitre était focalisée sur l'intégration de champs raster résultant d'une méthode d'interpolation bien particulière : l'estimation de la densité par la méthode des noyaux (champs KDE¹⁹⁵). Ce choix était d'abord motivé par la popularité de la méthode

¹⁹⁵ « Kernel Density Estimation ».

dans le domaine de la cartographie criminelle pour produire des cartes de hotspots (deuxième objectif de recherche). D'un point de vue plus théorique, KDE est une méthode d'interpolation spatiale originale car elle produit une vision spatialement continue de phénomènes discrets (tels que les délits). Le choix de cette méthode s'avérerait donc pertinent pour valider l'intégration de champs dans notre modèle, mais surtout pour apporter un type d'interface cartographique original dans le domaine du SOLAP appliqué sur des phénomènes spatialement discrets. Ce nouveau support visuel des données spatiales a ainsi l'avantage de minimiser l'impact du problème des unités spatiales modifiables (MAUP¹⁹⁶) puisque les frontières d'un hotspot résultant d'un KDE sont définies à partir d'un critère de distance qui concerne toutes les données situées à proximité (dans la limite du rayon de la fenêtre de lissage).

Pour que des champs KDE puissent alimenter un SOLAP, nous avons émis l'hypothèse que pour chaque opération impliquant une agrégation de ces champs, l'utilisateur devrait avoir l'illusion de calculer un champ KDE sur base d'un nuage de points défini par les paramètres de la dite opération (coupes et forages). L'outil SOLAP pourrait alors être considéré comme une interface rapide, conviviale et élégante pour la production de cartes de hotspots lorsqu'elle intègre des champs KDE (agrégation des champs est globalement plus rapide que calculer des KDE). De plus, ces hotspots pourraient faire l'objet d'une analyse multidimensionnelle à travers les interfaces graphiques ou tabulaires. Nous avons ensuite démontré que pour que cette hypothèse soit vraie, une mesure ROLAP « champ KDE » doit être contrainte par deux métadonnées spatiales supplémentaires : la fonction KDE et la taille de la fenêtre de lissage.

A partir d'un exemple d'hypercube raster KDE intégrant des données criminelles de la ville de Londres, nous avons illustré plusieurs opérations SOLAP impliquant une mesure « densité de délits » (mesure KDE), mais aussi une mesure « nombre de délits ». Il est en effet important pour l'utilisateur de manipuler ces deux mesures afin d'alimenter l'interface cartographique par des hotspots, tout en ayant accès aux quantités de délits absolues à l'intérieur de ces hotspots via l'interface graphique (ou tabulaire). Parmi ces opérations, celles appliquées sur l'interface cartographique (affichant les dimensions spatiales) sont évidemment les plus intéressantes pour les champs KDE. Il est ainsi possible d'exploiter des coupes non-spatiales pour définir les données de la carte produite, des coupes spatiales pour isoler certaines zones d'intérêt, ou encore des comparaisons cartographiques efficaces. Le forage spatial sur champs KDE est particulièrement intéressant lorsqu'il est appliqué entre des niveaux spatiaux définis par une fenêtre de lissage et/ou une fonction KDE différente(s). En effet, une modification de ces paramètres entraîne une généralisation cartographique que l'utilisateur peut adapter en temps réel à son échelle d'analyse : un « drill down » spatial produit une carte où les hotspots sont plus petits et plus nombreux, et donc plus adaptée à une analyse plus locale (plus grande échelle), et *vice versa* pour le « roll up »¹⁹⁷. Ce changement d'échelle en temps réel dans un SOLAP KDE constitue une solution potentielle à l'ambiguïté du choix de la fonction KDE et de la fenêtre de lissage pour la production de carte de hotspots, en particulier dans le domaine du « crime mapping ».

¹⁹⁶ « Modifiable Areal Unit Problem ».

¹⁹⁷ « Drill down » et « roll up » sont des opérations SOLAP de forage consistant respectivement à descendre ou monter dans une hiérarchie dimensionnelle (en l'occurrence, la hiérarchie spatiale).

Le chapitre 7 a clairement démontré l'intérêt d'un SOLAP raster pour l'analyse de phénomènes spatialement continus, mais aussi discrets. La **troisième partie** de la thèse, composée du **chapitre 8** et de cette conclusion, était d'abord consacrée à la validation de notre modèle au moyen d'une architecture web complètement « open source », notamment sur un hypercube de données impliquant davantage de dimensions non-spatiales et de niveaux spatiaux. En plus de valider le modèle, ce prototype englobait notre deuxième objectif de recherche : la conception d'un outil SOLAP adapté aux besoins de la police. En fin de chapitre, une étude comparative nous a permis de mieux situer les places respectives du SOLAP raster et du SOLAP vectoriel. Trois perspectives intéressantes ont ensuite été énoncées par rapport l'approche SOLAP raster. Les derniers paragraphes de cette synthèse décrivent ce dernier chapitre.

Notre prototype exploite essentiellement les fonctionnalités raster du SGBD PostGIS. Le rôle du serveur SOLAP programmé en aval de l'entrepôt étant essentiellement la construction des requêtes SQL transmises à l'entrepôt de données raster, cet outil rentre dans la catégorie des « SIG dominants ». Malgré l'absence de véritable serveur SOLAP (par exemple : GeoMondrian), la richesse des opérations SOLAP atteintes uniquement à travers les fonctionnalités raster de l'entrepôt spatial confirme bien notre hypothèse de départ : puisque la structure d'un raster se rapproche de celle d'un cube de données (plus précisément, un cube MOLAP), un outil SOLAP original pourrait être implémenté en exploitant uniquement les fonctionnalités propres au domaine SIG raster (notamment l'algèbre de carte). Ce type d'outil est cependant très différent des SOLAP vectoriels puisqu'il intègre les dimensions spatiales *X*, *Y* directement dans sa structure multidimensionnelle. Cette particularité permet notamment de gérer les phénomènes spatialement continus, mais aussi d'autoriser l'importation à la volée de membres spatialisés au moment de l'analyse. Ces aspects importants ont pu être validés par les trois hypercubes construits et exploités par le prototype : London (hotspots de criminalité basé sur Londres), Seattle (hotspots de criminalité basés sur Seattle) et Moon (température à la surface de la Lune). Alors que les deux premiers hypercubes concernent notre objectif « crime mapping », le cube « Moon » démontre que l'outil peut également intégrer des données spatialement continues dans d'autres domaines.

Au niveau applicatif, le prototype est capable de gérer l'analyse de hotspots de criminalité à travers la modélisation SOLAP raster KDE (voir chapitre 7). Notre serveur SOLAP construit automatiquement des magasins de données (hypercubes raster en aval d'un ED vectoriel) qui stockent les champs KDE. Ces hypercubes raster permettent à l'utilisateur d'analyser les hotspots de criminalité, selon diverses échelles d'analyse préprogrammées, de manière rapide et conviviale. De plus, l'emploi de fonctions d'agrégations holistiques telles que l'étendue dans l'agrégation des champs KDE permet la génération de cartes originales que l'on pourrait qualifier de « cartes de hotspots de variation ». En exploitant les fonctionnalités de l'interface graphique, l'utilisateur a également la possibilité d'approfondir l'analyse des hotspots intéressants révélés par ces cartes, à travers leurs dimensions non-spatiales (type délit, jour de la semaine, mois, *etc.*) et leur mesure « nombre de délits ».

Après avoir présenté notre prototype SOLAP raster, une étude comparative a été dressée afin de mieux cerner la place relative du raster dans le SOLAP par rapport au vecteur. Cette étude comparait trois types de modélisations SOLAP maillées sur des données discrètes : SOLAP raster, SOLAP vectoriel et SIG. L'approche SOLAP

vectorielle maillée (Mc Hugh, 2008) est avantageuse car elle peut être implémentée facilement dans un SOLAP vectoriel et permet l'accès aux données discrètes individuelles, ce qui est impossible avec l'approche raster développée dans cette thèse. L'approche SIG est une approche purement transactionnelle et se montre aussi efficace que le SOLAP uniquement avec de petits jeux de données. Les résultats de l'étude ont montré que l'approche raster affiche de meilleures performances que les deux approches vectorielles (SOLAP et SIG) lorsque la densité moyenne de points de l'ED vectoriel par pixel de l'ED raster est supérieure à une certaine valeur seuil (dans notre cas : de l'ordre de 0,1). Comme nous l'avons montré, la densité moyenne de points par pixel d'un magasin de données raster diminue lorsque le nombre de membres (spatiaux ou non) et/ou le nombre de dimensions diminue(nt). Dans le cas de données discrètes, le SOLAP vectoriel est donc clairement plus adapté aux données plus détaillées (hypercube à faible densité), et le SOLAP raster est plus adapté aux données plus agrégées (hypercube à forte densité). Cette étude confirme donc la place du SOLAP raster comme magasin de données en aval de l'ED vectoriel pour une analyse de données discrètes (à l'instar du MOLAP en aval du ROLAP). L'étude comparative nous a également permis de confirmer l'hypothèse que nous avons émise concernant la performance des agrégations spatiales avec l'approche raster qui se révèlent nettement plus efficaces qu'avec une approche vectorielle. Ce gain de performance du raster par rapport au vecteur s'explique de la même manière que le gain de performance du MOLAP par rapport au ROLAP : dans le contexte de l'OLAP, l'indexation sur des tableaux est plus efficace que l'indexation sur des tables relationnelles. Ce genre de résultat encourage d'ailleurs l'exploitation de la bande des raster comme dimension, approche qui n'a pas été implémentée dans le prototype.

Finalement, trois perspectives de recherche ont été abordées à la fin de la thèse. Tout d'abord, nous avons vu que le tuilage d'un hypercube raster pouvait partiellement résoudre le problème de la gestion de la faible densité des dimensions raster. Un schéma en constellation a été proposé comme piste de recherche pour exploiter ces tuiles sous PostGIS. Comme nous l'avons vu, le modèle de tuilage dans le SGBD Oracle présente nettement moins de redondance que celui de PostGIS et par conséquent, l'implémentation de notre modèle d'hypercube raster tuilé serait certainement plus performante avec Oracle GeoRaster. Comme démontré par Li *et al* (2014), le tuilage peut aussi être exploité pour distribuer les requêtes SOLAP raster sur plusieurs serveurs afin de réduire considérablement les temps de réponse. En deuxième perspective, nous avons vu que cette méthode d'optimisation serait tout aussi valable en divisant les requêtes selon une dimension ROLAP d'un hypercube raster (tuilé ou non). Enfin, notre troisième et dernière perspective concernait l'exploitation de cubes raster dimensionnels (à ne pas confondre avec un cube raster de mesures) pour intégrer des entités géographiques définies à la fois dans l'espace et dans le temps (intégration à la volée dans l'analyse de l'utilisateur). De cette manière, un hypercube raster apporterait une solution élégante à la problématique du SOLAP en fonction d'entités géographiques mobiles et/ou à géométrie variable. Un exemple a été fourni notamment dans le domaine de la climatologie.

Contribution au domaine SOLAP

Pour remplir notre objectif visant à intégrer la continuité spatiale au SOLAP, nous avons développé un modèle d'hypercube exploitant le format raster. Cette intégration du raster n'a cependant pas été aisée puisqu'il a fallu complètement repenser la manière de concevoir le SOLAP depuis ses origines (Bédard, 1997). En effet, en observant les différents schémas SOLAP qui peuplent la littérature, on se rend compte que même au niveau purement conceptuel, l'approche SOLAP classique est très étroitement liée au vecteur. De manière générale, l'idée est d'associer des géométries aux membres de dimensions sémantiques (non-spatiaux) ou aux mesures d'un schéma en étoile OLAP. La structure multidimensionnelle de ce type d'outil est donc basée avant tout sur l'aspect sémantique des dimensions, alors que la spatialité, déduite de ces géométries, est rejetée en dehors de la structure multidimensionnelle de l'entrepôt. L'approche raster que nous avons développée est complètement différente puisque les dimensions spatiales planimétriques (X , Y) sont considérées comme des branches du schéma en étoile conceptuel, et font donc directement partie de la structure multidimensionnelle de l'ED. Dans cette approche alternative, l'interaction entre le spatial et l'OLAP est ainsi beaucoup plus forte que dans l'approche vectorielle classique. Grâce à cette interaction, un SOLAP raster est caractérisé par une grande souplesse spatiale puisque n'importe quelle entité définie dans l'espace (membre géographique) peut être importée à la volée dans l'analyse de l'utilisateur. Il est même théoriquement possible d'appliquer des opérations SOLAP sur des entités géographiques caractérisées par des hiérarchies complexes (par exemple : entités mobiles).

Dans un SOLAP raster, les membres spatiaux sont indexés implicitement grâce à leurs coordonnées dans l'hypercube de données, alors que dans un SOLAP vectoriel, les géométries sont mises en relation avec des membres sémantiques de l'ED grâce à des pointeurs. Cette différence est très similaire à la distinction entre cubes MOLAP (basés sur des tableaux multidimensionnels) et cubes ROLAP (basés sur des relations) dans la littérature purement OLAP. Comme nous l'avons démontré au cours de cette thèse, les SOLAP raster et vectoriels sont d'ailleurs caractérisés respectivement de la même manière que les cubes MOLAP et ROLAP. Les agrégations sur des dimensions vectorielles (ROLAP) sont moins efficaces que sur des dimensions raster (MOLAP). Comme dans le ROLAP, un SOLAP vectoriel est adapté aux cubes de faible densité et comme dans le MOLAP, un SOLAP raster est adapté aux cubes de forte densité. Pour exploiter les points forts de ces deux approches, la littérature purement OLAP propose une architecture hybride (HOLAP) où des magasins de données (MOLAP), stockant des faits plus agrégés, sont construits sur base d'un ED ROLAP, stockant des faits plus détaillés. Nous avons également démontré qu'une architecture hybride SOLAP vecteur/raster exploite les avantages des deux approches spatiales de la même manière.

Pour minimiser le problème de faible densité des hypercubes raster, nous avons proposé une solution consistant à associer les dimensions non-spatiales de l'ED à des dimensions ROLAP. Ainsi, la faible densité est gérée sur ces dimensions, mais pas sur les dimensions raster. Une perspective de recherche a cependant été énoncée pour minimiser la faible densité des dimensions raster en exploitant le concept de tuilage. Une récente recherche propose même de distribuer les requêtes SOLAP sur des tuiles raster vers un « cloud » d'entrepôts (Li *et al*, 2014).

L'intégration de la continuité spatiale dans le SOLAP avait déjà été abordée dans plusieurs travaux (Ahmed & Miquel, 2005; Vaisman & Zimányi, 2009; Bimonte & Kang, 2010; Zaamoune *et al*, 2013), essentiellement sur base de l'approche vectorielle. Dans ces approches, l'interpolation des champs est effectuée à la volée au moment de la requête. L'utilisateur est donc libre de choisir sa méthode d'interpolation au moment de son analyse. Cependant, des techniques d'interpolation sophistiquées impliqueraient un temps de réponse du système relativement lent. C'est par exemple le cas de KDE, lorsque le champ est calculé sur base d'un nuage de points dense et/ou sur une grille comportant un grand nombre de mailles. Dans notre approche SOLAP raster, tous les champs sont calculés au moment de l'intégration des données, ce qui garantit un temps de réponse plus rapide, mais moins de liberté dans le choix de la méthode d'interpolation au moment de l'analyse.

Enfin, la plupart des travaux SOLAP traitant des données continues se sont naturellement focalisés sur l'étude des phénomènes spatialement continus (température, activités sismiques, *etc.*). A notre connaissance, aucune recherche avant nous ne s'était penchée sur la question de la visualisation spatialement continue de phénomènes discrets. L'intégration des champs KDE dans notre SOLAP raster a ainsi démontré qu'il était possible de réduire l'impact du MAUP (« Modifiable Areal Unit Problem ») dans un SOLAP. L'approche SOLAP raster a donc un potentiel très intéressant pour modéliser les phénomènes continus (*cf.* cube Moon), mais également les phénomènes discrets (*cf.* hypercubes London et Seattle).

Contribution au domaine « crime mapping »

L'outil SOLAP développé dans ce travail avait pour but d'enrichir l'analyse de hotspots de criminalité par KDE, technique très populaire en cartographie criminelle. Ainsi, notre prototype est capable de générer des cartes de hotspots à travers des opérations SOLAP sur les dimensions décrivant les données, qu'elles soient spatiales ou non (type de délit, jour de la semaine, mois, saison, heure, *etc.*). Une fois les données intégrées, la production des cartes s'exécute rapidement, conformément aux exigences attendues d'un outil SOLAP (agréger des champs KDE pré-calculés est globalement plus rapide que construire les champs à la volée).

Notre outil ne se limite pas à un simple générateur de cartes. En effet, la richesse du SOLAP autorise une analyse plus approfondie des hotspots grâce à la visualisation interactive des dimensions non-spatiales dans l'interface graphique. Afin de guider l'utilisateur dans l'exploration de ses données, il est par exemple possible d'analyser l'évolution temporelle des hotspots, la répartition des types de délits qu'ils renferment, ou même les deux à la fois. Des comparaisons peuvent être effectuées dans cet espace multidimensionnel de manière souple, aussi bien à travers des cartes que des graphiques. Grâce à l'emploi de fonctions d'agrégation holistiques (écart-type, étendue), il est également possible de délimiter et d'isoler automatiquement les zones spatiales présentant des anomalies (brusques changements du nombre de délits à travers le temps).

Enfin, le SOLAP raster apporte une solution partielle à l'ambiguïté du paramétrage des fonctions KDE relevée dans la littérature « crime mapping » (Chainey, 2013). En effet, ces paramètres dépendent avant tout de l'échelle d'analyse de l'utilisateur. Puisque le pré-calcul des niveaux spatiaux dans l'ED s'opère à travers plusieurs combinaisons de

résolution de l'hypercube raster et de taille de fenêtre de lissage, les opérations de forage spatial permettent à l'utilisateur de comparer rapidement plusieurs échelles d'analyse avant d'approfondir celle qu'il juge adéquate.

Perspectives de recherche

Le prototype développé dans ce travail est assez rudimentaire et pourrait être enrichi par d'autres fonctionnalités. Par exemple, l'importation de membres géographiques au moment de l'analyse, qui est un atout important de l'approche SOLAP raster, n'est possible que par digitalisation¹⁹⁸. Même si, dans le cadre la thèse, cette digitalisation suffisait à prouver la flexibilité de l'outil pour la gestion ce type de membre, l'importation de fichiers spatiaux (par exemple : shapefiles), comme couche composée de plusieurs membres géographiques, constituerait une fonctionnalité particulièrement puissante pour un SOLAP. De plus, les mesures agrégées en fonction de ces membres ne peuvent être visualisées que dans l'interface graphique, et non dans l'interface cartographique. Comme nous l'avons déjà signalé, les membres spatialisés pourraient également être associés à un cube ou à un hypercube dimensionnel, afin d'importer des entités géographiques caractérisées par une hiérarchie complexe (par exemple : entités mobiles).

Le prototype souffre encore de problèmes d'ergonomie. Les opérations de forage spatial nécessitent une certaine prudence de la part de l'utilisateur afin de ne pas exécuter des calculs d'agrégation trop lourds pour l'OLAP. En effet, il est recommandé de toujours combiner les « spatial drill down » (augmentation de la résolution) à des coupes spatiales afin de réduire la taille des raster agrégés. Idéalement, cette opération pourrait systématiquement impliquer une coupe sur une fenêtre spatiale de taille raster fixe (par exemple : 500 * 500 pixels) de manière transparente du point de vue l'utilisateur. En un seul clic, ce dernier appliquerait un « zoom intelligent » où le temps de réponse du système resterait stable, peu importe l'échelle d'analyse demandée.

La possibilité de produire des cartes de hotspots pondérées en fonction du type de délit a été démontrée d'un point de vue théorique. Néanmoins, son implémentation dans PostGIS nous a confronté à des problèmes techniques relativement difficiles à surmonter (mais certainement pas impossibles).

Le choix de PostGIS raster pour l'ED était avant tout motivé par la volonté de nous limiter uniquement à des outils « open source ». Cette philosophie nous semblait logique pour rester dans la continuité des travaux de notre prédécesseur à l'Unité de Géomatique (Laplanche, 2006). Cependant, un outil SOLAP raster développé dans Oracle ouvrirait certainement la porte vers des perspectives intéressantes : notamment une modélisation continue du temps à travers la troisième dimension raster (*B*), ou un tuilage efficace pour la gestion de la faible densité des dimensions raster.

Comme nous l'avons déjà signalé, pour implémenter notre modèle, nous avons opté pour une approche « SIG dominant ». Puisque cette thèse s'inscrit avant tout dans le domaine de la géomatique, et que l'idée générale était de faire le rapprochement entre

¹⁹⁸ Dans cette thèse, le terme « digitalisation » fait référence à la construction d'entités vectorielles en saisissant leurs coordonnées à l'écran.

l'OLAP et le format raster, cette approche nous est apparue comme la plus évidente. De plus, nous avons vu que l'approche SOLAP raster était conceptuellement très différente de l'approche SOLAP vectorielle qui caractérise les outils SOLAP actuels : GeoMondrian (Spatialytics, 2014), Map4Decision (Intelli³, 2010), *etc.* Il nous a donc semblé plus simple de développer notre propre serveur SOLAP en exploitant directement les fonctionnalités de PostGIS plutôt que de nous baser sur un de ces outils. Il faut cependant remarquer qu'avec notre approche, l'emploi d'un de ces serveurs SOLAP vectoriels ne serait certainement pas exclue dans une architecture hybride (vecteur/raster).

Dans de futurs travaux, il serait très intéressant de développer notre modèle SOLAP raster selon une approche « MOLAP dominant ». En raison des ressemblances évidentes entre le raster et le MOLAP, ce type de développement consisterait à implémenter les fonctionnalités SIG directement sur un outil purement MOLAP tel que l'outil « open source » développé par MacroFocus (2015). Sur leur site internet, l'agrégation sur des objets spatiaux fait d'ailleurs partie de la liste des perspectives (ainsi que la distribution des requêtes).

Finalement, l'intégration du raster dans le SOLAP ouvre la voie vers bien d'autres domaines nécessitant une vision continue de l'espace : télédétection, environnement, climatologie, *etc.* Puisque l'objectif de ce travail était focalisé sur le « crime mapping », ces perspectives d'application restent ouvertes. De plus, le modèle d'entrepôt raster, initialement pensé pour le SOLAP, constituerait une base intéressante pour alimenter l'autre grande branche de l'informatique décisionnelle : la fouille de données automatisée, ou « data mining ».

Bibliographie

- Abello, A., Samos, J. & Saltor, F. (2002). On relationships offering new drill-across possibilities. *Proceedings of the Fifth International Workshop on Data Warehousing and OLAP*. New-York : ACM, 7-13.
- Apache Software Foundation (2015). *Welcome to the Apache Software Foundation*. <https://www.apache.org/>. Consulté le 03/01/2015.
- Adelson, E. H., Anderson, C. H., Bergen, J. R. , Burt, P. J. & Ogden, J. M. (1984). Pyramid methods in image processing, *RCA Engineer*, 33-41.
- Ahmed, T. O. & Miquel, M. (2005). Multidimensional structures dedicated to continuous spatiotemporal phenomena. *Proceedings of the Twenty-Second British National Conference on Databases: Enterprise, skills and innovation*. Sunderland : Springer-Verlag, 29-40.
- Ahmed, T. O. (2008). Continuous spatial data warehousing. *Proceedings of the Ninth International Arab Conference on Information Technology*.
- Badard, T., Dube, E., Diallo, B., Mathieu, J. & Ouattara, M. (2009). Open source geospatial Business Intelligence (BI) in action. *Proceedings of the Tenth International Opensource Geospatial Research Symposium*. Nantes.
- Bédard, Y. (1997). Spatial OLAP. *Forum annuel sur la R-D, Géomatique VI: Un monde accessible*. 13-14 novembre. Montréal.
- Bédard, Y., Proulx, M. J. & Rivest, S. (2005). Enrichissement du OLAP pour l'analyse géographique : exemples de réalisation et différentes possibilités technologiques. *1ère journée francophone sur les Entrepôts de Données et l'Analyse en ligne (EDA 2005)*. Lyon.
- Bédard, Y., Bernier E., Larrivée S., Nadeau M., Proulx M. J. & Rivest S. (2009). *Spatial OLAP portal*. <http://spatialolap.scg.ulaval.ca/>. Consulté le 07/01/2015.
- Bédard, Y. (2010). Le géodécisionnel : Origine, évolution, état de l'art, enjeux, R&D. École Nationale Supérieure des Mines de Paris - Centre de recherche sur les Risques et les Crises. 29 octobre. Sophia-Antipolis.
- Bédard, Y. (2014). Beyond GIS: Spatial Online Analytical Processing and Big Data. *The 2014 Dangermond Lecture*. Santa-Barbara.
- Bertin, J. (1967). *Sémiologie graphique : Les diagrammes, les réseaux, les cartes*. Paris, La Haye, Mouton, Gauthier-Villars.
- Bimonte, S. (2007). *Intégration de l'information géographique dans les entrepôts de données et l'analyse en ligne : de la modélisation à la visualisation*. Institut National des Sciences Appliquées de Lyon, Lyon.
- Bimonte, S. & Kang, M. A. (2010). Towards a model for the multidimensional analysis of field data. In B., Catania, M., Ivanovic & B., Thalheim (Eds.), *Proceedings of the Fourteenth east European conference on Advances in databases and information systems*. Berlin : Springer-Verlag, 58-72.

- Bimonte S., Kang M. A., Paolino L., Sebillio M., Zaamoune M. & Vitiello, G. (2014). OLAPing Field Data: a Theoretical and Implementation Framework. *Fundamenta Informaticae - Warehousing and OLAPing Complex, Spatial and Spatio-Temporal Data*, 132(2), 267-290.
- Cabibbo, L. & Torlone, R. (1998). A Logical Approach to Multidimensional Databases. *Proceedings of the Sixth International Conference on Extending Database Technology*. Spain : Springer-Verlag, 183-197.
- Chainey, S. P., Tompson, L. & Uhlig, S. (2008). The utility of hotspot mapping for predicting spatial patterns of crime. *Security Journal*, 21, 4-28.
- Chainey, S. (2013). Examining the influence of cell size and bandwidth size on kernel density estimation crime hotspot maps for predicting spatial patterns of crime. *Bulletin of the Geographical Society of Liege*, 60, 7-19.
- Chee, T., Chan, L. K., Chuah, M. H., Tan, C. S., Wong, S. F. & Yeoh, W. (2009). Business Intelligence systems: state-of-the-art review and contemporary applications. *Symposium on Progress in Information & Communication Technology*. Kuala Lumpur.
- Chrisment, C., Pujolle, G., Ravat, F., Teste, O. & Zurfluh, G. (2005). Entrepôts de données. *Techniques de l'Ingénieur Bases de données*, H3870, TIB309DUO.
- Codd, E. F. (1970). A relational model of data for large shared data banks. *Communications of the ACM*, 13(6), 377-387.
- Codd, E.F., Codd, S.B. & Salley, C.T. (1993). *Providing OLAP (On-line Analytical Processing) to User-Analysts: An IT Mandate*. E.F Codd & Associates.
- Collet, C. (1992). *Systèmes d'information géographique en mode image*. Lausanne : Presses polytechniques et universitaires romandes.
- Dabin, D., Dickens, C. & Wouters P. (2013). Estimateur à noyau sur réseaux : Une application aux accidents de la route belges. *Bulletin de la société géographique de Liège*, 60, 21-31.
- Dang T. N. U. (2012). *PostGIS Raster: Extending postgresSQL for the support of continuous fields*. Université Libre de Bruxelles, Bruxelles.
- Dark, S. J. & Bram, D. (2007). The modifiable areal unit problem in physical geography. *Progress in Physical Geography*, 31(5), 471-479.
- Davis, J. C. (1986). *Statistics and data analysis in geology*, 2nd ed. New York: John Wiley & Sons.
- Demers, M. N. (2001). *GIS Modeling in Raster*. New York : John Wiley & Sons.
- Denègre, J. (2005). *Sémiologie et conception cartographique*. Paris : Lavoisier.
- Devlin, B. A., Murphy, P. T. (1988). An architecture for a business and information system. *IBM Systems Journal*, 27(1), 60-80.
- DHTMLX (2015). *Javascript UI library, ajax components & HTML5 framework*. <http://dhtmlx.com/>. Consulté le 03/04/2015.
- De Smith, M., J., Goodchild, M., F. & Longley, P., A. (2007). *Geospatial Analysis – A Comprehensive Guide to Principles, Techniques and Software Tools*. Leicester: Matador.

- Di Salvo M., Gadais, M. & Roche-Woillez, G. (2005). *L'estimation de la densité par la méthode du noyau: méthodes et outils*. Lyon : Certu.
- Donnay, J. P. & Binard, M. (1991). Spatially continuous processing within a raster based GIS: some examples of geographical models. *Bulletin de la Société belge de Géographie*, 2, 233-244.
- Donnay, J. P. & Unwin, D. (2001). Modelling Geographical Distributions in Urban Areas. In J.-P., Donnay, M. J., Barnsley & P. A., Londley (Eds.), *Remote Sensing and Urban Analysis*. London et New York : Taylor & Francis, 205-224.
- Donnay, J. P. (2005a). Formalisation des informations géographiques en mode maillé. *Revue internationale de géomatique*, 15 (4), 415-438.
- Donnay, J. P. (2005b). Les systèmes d'information géographiques (SIG). Préliminaires à un usage dans l'enseignement. *Bulletin de la Société géographique de Liège*, 45, 45-52.
- Donnay, J. P. (2010). *Cartographie mathématique*. Université de Liège, Liège.
- Donnay, J. P. (2013). *Guide de rédaction des cartes thématiques. Méthodes et consignes*. Liège : Université de Liège.
- Eastman, J. R., Jin, W, Kyem, P. A. K. & Toledano, J. (1995). Raster procedures for multi-criteria/multi-objective decisions. *Photogrammetric Engineering & Remote Sensing*, 61(5), 539-547.
- Eastman, J. R. (2009). *IDRISI Taiga Guide to GIS and Image Processing*. Worcester : Clark Labs.
- Eck, J. E., Chainey, S. P., Cameron, J. G., Leitner, M. & Wilson, R. E. (2005). *Mapping crime: Understanding hot spots*. Washington: National Institute of Justice.
- Egenhofer, M. (1989). A formal definition of Binary Topological Relationships. *Proceedings of the Third International Conference on Foundations of Data Organization and Algorithms*. Paris: Springer-Verlag.
- El-Gamal, N., Galal-Edeen, G. & El-Bastawissy, A. (2005). Towards a Generic Conceptual Model for Data Warehouses. *Proceedings of the Fifth International Business Information Management Association Conference*. Egypt: IBIMA, 262-269.
- El-Gamal N. (2007). Data warehouse conceptual modeling approaches. In M. H., Elwany & A. B., Eltawil (Eds.), *Proceedings of the Thirty-Seventh International Conference on Computers and Industrial Engineering*. Alexandria.
- Ervin, D. (2015). *MAUP – An Introduction to the Modifiable Areal Unit Problem*. <http://gispopsci.org/maup/> . Consulté le 12/05/15
- ESRI (2012). *Desktop help 10.0 : Bienvenue dans la bibliothèque d'aide ArcGIS*. <http://help.arcgis.com/fr/arcgisdesktop/10.0/help/>. Consulté le 05/01/2015.
- Fiume, E. L. (1989). *The Mathematical Structure of Raster Graphics*. Boston : Academic Press.
- Franke, R. (1982). Smooth Interpolation of Scattered Data by Local Thin Plate Splines. *Computer and Mathematics with Applications*, 8 (4), 273-281.
- Franklin, C. (1992). An introduction to geographic information systems: Linking maps to databases. *Database*, 15 (2), 13-21.

- Golfarelli, M., Maio, D. & Rizzi, S. (1998). The Dimensional Fact Model: A Conceptual Model for Data Warehouses. *International Journal of Cooperative Information Systems*, 7(2-3), 215-247.
- Gomez, L., Kuijpers B., Vaisman, A (2008). Aggregation languages for moving object and places of interest. *Proceedings of the 2008 ACM symposium on Applied computing (SAC '08)*. New York: ACM. 857-862.
- Gomez, L., Vaisman, A. & Zimanyi, E. (2010). Physical design and implementation of spatial data warehouses supporting continuous fields. In T. B., Pedersen, M. K, Mohania, & A. M., Tjoa (Eds.), *Proceedings of the Twelfth International Conference on Data Warehousing and Knowledge Discovery*. Berlin : Springer-Verlag.
- Gomez, L.I., Gomez, S.A. & Vaisman, A. (2011). Analyzing continuous fields with OLAP cubes. *Proceedings of the 14th international workshop on Data Warehousing and OLAP*. Glasgow : ACM.
- Gomez, L.I., Gomez, S.A & Vaisman, A. (2012). A generic data model and query language for spatiotemporal OLAP cube analysis. *Proceedings of the Fifteenth International Conference on Extending Database Technology*. Berlin : ACM, 300-311.
- Grass GIS (2015). *Grass GIS home*. <http://grass.osgeo.org/>. Consulté le 12/03/2015.
- Gutierrez, A. & Baumann, P. (2008). Computing Aggregate Queries in Raster Image Databases Using Pre-Aggregated Data. *Proceedings of the World Congress on Engineering and Computer Science*. San Francisco.
- Hallo, P. (2012). L'identité à travers l'espace et le temps. *Vers une définition de l'identité et des relations spatio-temporelles entre objets géographiques*. Université de Liège, Liège.
- Han, J., Stefanovic, N. & Koperski, K. (1998). Selective materialization: an efficient method for spatial data cube construction. *Proceedings of the Second Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Melbourne : Springer-Verlag, 144-158.
- Harinarayan, V., Rajaraman, A. & Ullman, J. D. (1996). Implementing Data Cubes Efficiently. *ACM SIGMOD International Conference on Management of Data*. Montreal : ACM, 205-216.
- Hart, T. & Zandbergen, P. (2014). Kernel density estimation and hotspot mapping: Examining the influence of interpolation method, grid cell size, and bandwidth on crime forecasting. *Policing: An International Journal of Police Strategies & Management*, 37 (2), 305-323.
- Intelli³ (2010). *Intelli³*. <http://www.intelli3.com>. Consulté le 08/01/2014.
- ISO 19123 (2005). *Information géographique - Schéma de la géométrie et des fonctions de couverture*.
- ISO 19128 (2005). *Geographic Information - Web map server interface*.
- Jagadish, H.V. (1990). Linear clustering of objects with multiple attributes. *Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data*. Atlantic City : ACM, 332-342.
- Kang M.A. (2001). *Indexation et visualisation des données de champs continus dans les SIG*. Institut National des Sciences Appliquées de Lyon, Lyon.

- Kang, M. A., Zaamoune M., Pinet F., Bimonte S. & Beaune P. (2013). Optimisation des performances des opérations d'agrégation au sein des entrepôts de grilles spatialisées. *Conférence Internationale de Géomatique et d'analyse spatiale (SAGEO 2013)*. Brest.
- Kasprzyk J. P., Trotta M., Broxham K. & Donnay J. P. (2012). Reconstitution of the journeys to crime and location of their origin in the context of a crime series. A raster solution for a real case study. In M., Leitner (Ed.) *Crime Modeling and Mapping Using Geospatial Technologies*. New York: Springer-Verlag, 125-143.
- Kimball, R. (1996). *The Data Warehouse Toolkit: Practical Techniques for Building Dimensional Data Warehouses*. New York : John Wiley & Sons.
- Kimball, R. & Ross M. (2013). *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling, Third Edition*. New York : John Wiley & Sons.
- Kothuri R. K. V., Ravada S. & Abugov D. (2002). Quandtree and r-tree indexes in Oracle Spatial: a comparison using gis data. *Proceedings of ACM GISMOD Conference*, New York : ACM, 546-557.
- Kuijpers, B., Vaisman, A. (2007). A Data Model for Moving Objects Supporting Aggregation. *Proceedings of the 23rd International Conference on Data Engineering Workshops. ICDE 07*. 15-20 avril 2007. Istanbul, Turquie.
- Laplanche, F. (2006). *Environnement de conception de bases de données spatiales sur Internet*. Université de Liège, Liège.
- Laurini, R. & Servignes, S. (2008). Panorama des potentialités SIG en 3 dimensions : Vers des modèles virtuels 3D de villes. *Revue XYZ*, 11, 22-26.
- Ledoux, H. & Gold, C. M. (2006). A Voronoi-based Map Algebra. *Proceedings of the Twelfth International Symposium on Spatial Data Handling*. Heidelberg : Springer, 117-131.
- Levine, N. (2010). *CrimeStat: A Spatial Statistics Program for the Analysis of Crime Incident Locations (v 3.3)*. Ned Levine & Associates, Houston, TX, and the National Institute of Justice, Washington.
- Li, J., Meng, L., Wang F. Z., Zhang, W & Cai, W. (2014). A Map-Reduce-enabled SOLAP cube for large-scale remotely sensed data aggregation. *Computers & Geosciences*, 70, 110-119.
- Lieberwirth, U. (2008). 3D GIS Voxel-Based Model Building in Archaeology. In A., Posluschny, K., Lambers & I., Herzog (Eds.), *Proceedings of the 35th International Conference on Computer Applications and Quantitative Methods in Archaeology*, 10, Berlin, 8.
- Lin, W. Y. & Kuo, I. C. (2004). A Genetic Selection Algorithm for OLAP Data Cubes. *Knowledge and Information Systems*, 6(1), 83-102.
- Macrofocus (2015). *Multidimensional OLAP API by Macrofocus*. <http://www.molap.org> . Consulté le 17/05/2015.
- Malinowski, E. & Zimányi, E. (2004). Representing spatiality in a conceptual multidimensional model. In D., Pfoser, I.F., Cruz & M., Ronthaler. *Proceedings of the Twelfth ACM International Workshop on Geographic Information Systems*. Washington: ACM, 12-22.

- Malinowski, E. & Zimányi, E. (2006). Hierarchies in a multidimensional model: From conceptual modeling to logical representation. *Data & Knowledge Engineering*, 59(2), 348-377.
- Mapinfo (2015). *Mapxtreme for .net*. <http://www.mapinfo.com/product/mapxtreme-for-net/>. Consulté le 10/01/2015.
- Mapserver (2014). *Welcome to Mapserver: Mapserver 7.0.0 beta1 documentation*. <http://mapserver.org/>. Consulté le 03/04/15.
- Marchand, P., Brisebois, A., Bédard, Y. & Edwards, G. (2004). Implementation and evaluation of a hypercube-based method for spatio-temporal exploration and analysis, *Journal of the International Society of Photogrammetry and Remote Sensing*, 59(1-2), 6-20.
- McHugh, R., Rocje, S. & Bédard, Y. (2007). Potentiel de la structure matricielle pour optimiser l'analyse des données géodécisionnelles. *Congrès sur la gestion de l'information géospatale*. Québec.
- McHugh, R. (2008). *Intégration de la structure matricielle dans les cubes spatiaux*. Université Laval, Québec.
- Mena, J. (2003). *Investigative Data Mining for Security and Criminal Detection*. Burlington: Elsevier.
- Mennis, J., Viger, R. & Tomlin, C. D. (2005). Cubic Map Algebra functions for spatio-temporal analysis. *Cartography and Geographic Information Systems*, 30(1), 17-30.
- Microsoft (2014a). *Analysis services MOLAP performance Guide for SQL Server 2012 and 2014*. <http://msdn.microsoft.com/en-us/library/dn749781.aspx>. Consulté le 16/10/2014.
- Microsoft (2014b). *.Net downloads developer resources & case studies*. <http://www.microsoft.com/net>. Consulté le 10/01/2015.
- Microstrategy (2015). *Enterprise analytics, mobile & security*. <http://www.microstrategy.com/us/>. Consulté le 10/01/2015.
- Miquel, M., Bédard, Y. & Brisebois, A. (2002). Conception d'entrepôts de données géospaciales à partir de sources hétérogènes : Exemple d'application en foresterie. *Ingénierie des Systèmes d'Information*, 7(3), 89-111.
- Mitas, L. & Mitsova, H. (1999). *Geographical information systems: principles, techniques, management and applications*. New-York : John Wiley & Sons.
- Morain, S. & Baros, S. L. (1996). *Raster Imagery in Geographic Information Systems*. Santa Fe : OnWord Press.
- Open GIS Consortium (2006). *The OpenGIS Abstract Specification -Topic 6: Schema for coverage geometry and functions*, OpenGIS Project Document 99-049, Version 7.0.
- Oliver, M. A. & Webster, R. (2010). Kriging: a method of interpolation for geographical information systems. *International journal of geographical systems*, 4(3), 313-332.
- OpenLayers (2015). *Openlayers 3: Welcome*. <http://openlayers.org/>. Consulté le 03/04/15.
- Openshaw, S. (1983). *The modifiable areal unit problem*. Norwick (Norfolk), Geo Books.
- Open Street Map (2015). *Openstreetmap*. <http://www.openstreetmap.org>. Consulté le 10/02/15.

- Oracle (2003). *Georaster overview and concepts*.
https://docs.oracle.com/html/B10827_01/geor_intro.htm. Consulté le 05/01/2015.
- Oracle (2015). *BI: Business intelligence*. <http://www.oracle.com/us/solutions/business-analytics/businessintelligence/overview/index.html>. Consulté le 10/01/15
- Paige, D. A., Schofield, J. T., Calcutt, S. B., Avis, C., Mortensen, H. B. & Sullivan, M. T. (2011). LRO-L-DLRE-4-RDR-V1.0, NASA Planetary Data System.
- Pedersen, T. B., Jensen, C. S. & Dyreson, C. E. (2001). A foundation for capturing and querying complex multidimensional data. *Information Systems*, 26 (5), 383-423.
- Pendse, N. (2007). *What is OLAP?* Disponible sur <http://www.olapreport.com/fasmi.html>
- Pentaho (2014). *Pentaho community*. <http://community.pentaho.com/>. Consulté le 03/10/2014.
- Plante, M. (2014). *Vers des cubes matriciels supportant l'analyse spatiale à la volée dans un contexte décisionnel*. Université Laval, Québec.
- PostGIS (2015). *PostGIS 2.1.8 dev manual*. <http://postgis.net/docs/manual-2.1/>. Consulté le 08/04/2015.
- Proulx, M. J. & Bédard, Y. (2004). Le potentiel de l'approche multidimensionnelle pour l'analyse de données géospatiales en comparaison avec l'approche transactionnelle des SIG. *Colloque de Géomatique*. Montréal, Canada.
- Proulx, M. J. (2009). Veille technologique portant sur le mariage judicieux de l'intelligence d'affaires et de l'information géospatiale. *Colloque de Géomatique*. Montréal, Canada.
- QGIS, (2015). *Documentation*. <http://www2.qgis.org/fr/docs/index.html>. Consulté le 12/02/2015.
- Racine, P. & Cumming S. (2011). Stockage, manipulation et analyse de données matricielles avec PostGIS Raster. *Colloque de Géomatique*. Montréal, Canada.
- Regina, O. & Leo, S. (2011). *PostGIS in Action*. Stamford : Manning Publications Co.
- Riverst, S., Bédard, Y., Proulx, M. J., Nadeau, M., Hubert, F. & Pastor J. (2005). SOLAP: Merging Business Intelligence with Geospatial Technology for Interactive Spatio-Temporal Exploration and Analysis of Data, *ISPRS Journal of Photogrammetry & Remote Sensing*, 60, 17–33.
- Rivest, S., Bédard, Y., Proulx, M.J. & Nadeau, M. (2003). SOLAP: a new type of user interface to support spatio-temporal multidimensional data exploration and analysis. *Proceedings of the ISPRS Joint Workshop on Spatial, Temporal and Multi-Dimensional Data Modelling and Analysis*. Québec.
- SAP (2015). *Business Intelligence tools BI & analytics*. <http://www.sap.com/pc/analytics/business-intelligence/software/overview/index.html>. Consulté le 10/01/2014.
- Sapia, C., Blaschka, M., Höfling, G. & Dinter, B. (1999). Extending the E/R model for the multidimensional Paradigm. *Proceedings International Workshop on Data Warehouse and Data Mining in conjunction with the ER'98*. Germany: Springer, 105-116.
- Seattle Police (2015). *Seattle Open data*. <https://data.seattle.gov/>. Consulté le 03/04/2015.
- Shanmugasundaram, J., Fayyad, U. M. & Bradley, P. S. (1999). Compressed data cubes for OLAP aggregate query approximation on continuous dimensions. *Proceedings of the*

- Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York : ACM, 223-232.
- Sibson, R. (1981). A brief description of natural neighbor interpolation, chapter 2. In *Interpreting Multivariate Data*. New-York : John Wiley & Sons, 21-36.
- SpagoBI (2014). *Spagoworld: The open source initiative supported by engineering group*. www.spagoworld.org/. Consulté le 03/10/2014.
- Spatialytics (2014). *100% open source business intelligence*. <http://www.spagobi.org/>. Consulté le 03/10/2014.
- Trujillo, J., Palomar, M., Gomez, J. & Song, I. (2001). Designing Data Warehouses with OO Conceptual Models. *IEEE Computer*, 34 (12), 66-75.
- Taylor, C., Gorard, S. & Fitz, J. (2003). The modifiable areal unit problem: Segregation between schools and levels of analysis. *International Journal of Social Research Methodology*, 6(1), 41-60.
- Tomlin, C. D. (1983). A Map Algebra. *Proceedings Harvard Computer Graphics Conference*. Cambridge.
- Tomlin, C. D. (1990). *Geographic Information Systems and Cartographic Modeling*. Englewood Cliffs : Prentice-Hall.
- Tomlin, C. D. (1991). Cartographic Modeling. In D., Maguire, M.F., Goodchild & D. Rhind (Eds.), *Geographic Information Systems: Principles and Applications*. London : Longman, 361-374.
- Tranchant, M. (2013). *Couplage d'un HyperCube SOLAP et d'un outil de visualisation*. Ecole d'Ingénieurs Eicnam, Grenoble.
- Trotta, M. (2014). *Modelling serial offenders' spatial behaviours : new assumptions for geographic profiling*. Université de Liège, Liège.
- Tryfona, N., Busborg, F. & Christiansen, J. G. (1999). StarER: A Conceptual Model for Data Warehouse Design. *Proceedings of the Second International Workshop Data Warehousing and OLAP*. New-York : AMC.
- Tsois, A., Karayannidis, N. & Sellis, T. (2001). MAC: Conceptual Data Modeling for OLAP. *Proceedings of the International Workshop on design and Management of Data Warehouses*. Interlaken.
- UK Police (2014). *Crime Map*. <http://www.police.uk>. Consulté le 10/05/2014.
- Vaisman, A. & Zimanyi, E. (2009). A multidimensional model representing continuous fields in spatial data warehouses. *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. Seattle : ACM, 169-177.
- Vassiliadis, P. (1998). Modeling Multidimensional Databases, Cubes and Cube Operations. *Proceedings of the Tenth International Conference on Scientific and Statistical Database Management*. Capri.
- Wang, L., Jajodia, S., Wijesekera, D. (2007). *Preserving Privacy in On-Line Analytical Processing (OLAP)*. New-York: Springer.
- Workboys M. & Duckham M. (2004). *GIS: A computing Perspective* (2nd Ed.). USA: CRC Press.

- Zaamoune, M., Bimonte, S., Pinet, F., Beaune, P. (2013). Intégration des données champs continus incomplets dans l'OLAP : de la modélisation conceptuelle à l'implémentation. *9^e journées francophones sur les Entrepôts de Données et l'Analyse en ligne (EDA 2013)* Blois.
- Zaninetti J. M. (2005). *Statistique spatiale: méthodes et applications géomatiques*. Paris : Hermès Sciences Publications.
- Watson, D. F. & Philip, G. M. (1985). A refinement of inverse distance weighted interpolation. *Geo-Processing*, 2, 315- 327.
- Wikipédia (2015). *Théorie des cordes*. http://fr.wikipedia.org/wiki/Th%C3%A9orie_des_cordes. Consulté le 08/02/15.

ANNEXES

Annexe 1. Fichiers XML de définition des hypercubes raster

London_ok.xml

```
<tree id="fact_raster" data_table="crime_point2" data_geom="location_proj" fact_table="fact_raster">
  <item text="Trimester" id="trimester" niveau="all" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif">
    <item text="Winter" id="1" niveau="trimester" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif">
      <item text="January" id="2012-01-01" niveau="month" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
      <item text="February" id="2012-02-01" niveau="month" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
      <item text="March" id="2012-03-01" niveau="month" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
    </item>
    <item text="Spring" id="2" niveau="trimester" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif">
      <item text="April" id="2012-04-01" niveau="month" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
      <item text="May" id="2012-05-01" niveau="month" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
      <item text="June" id="2012-06-01" niveau="month" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
    </item>
    <item text="Summer" id="3" niveau="trimester" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif">
      <item text="July" id="2012-07-01" niveau="month" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
      <item text="August" id="2012-08-01" niveau="month" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
      <item text="September" id="2012-09-01" niveau="month" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
    </item>
    <item text="Fall" id="4" niveau="trimester" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif">
      <item text="October" id="2012-10-01" niveau="month" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
      <item text="November" id="2012-11-01" niveau="month" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
      <item text="December" id="2012-12-01" niveau="month" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
    </item>
  </item>
  <item text="Crime type" id="crime_type" niveau="all" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif">
    <item text="Robbery" id="Robbery" niveau="crime_type" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
    <item text="Shoplifting" id="Shoplifting" niveau="crime_type" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
    <item text="Criminal damage and arson" id="Criminal damage and arson" niveau="crime_type" im0="folderClosed.gif" im1="folderOpen.gif"
    im2="folderClosed.gif"/>
  </item>
</tree>
```

```

<item text="Burglary" id="Burglary" niveau="crime_type" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
<item text="Public disorder and weapons" id="Public disorder and weapons" niveau="crime_type" im0="folderClosed.gif" im1="folderOpen.gif"
im2="folderClosed.gif"/>
<item text="Drugs" id="Drugs" niveau="crime_type" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
<item text="Violent crime" id="Violent crime" niveau="crime_type" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
<item text="Other theft" id="Other theft" niveau="crime_type" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
<item text="Vehicle crime" id="Vehicle crime" niveau="crime_type" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
<item text="Anti-social behaviour" id="Anti-social behaviour" niveau="crime_type" im0="folderClosed.gif" im1="folderOpen.gif"
im2="folderClosed.gif"/>
<item text="Other crime" id="Other crime" niveau="crime_type" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
</item>
</tree>

```

Seattle_ok.xml

```
<tree id="seattle_fact_table" data_table="seattle_crime_point" data_geom="location" fact_table="seattle_fact_table">
  <item text="Year" id="year" niveau="all" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif">
    <item text="2013" id="2013" niveau="year" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif">
      <item text="Winter" id="t1" niveau="trimester" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif">
        <item text="January" id="01_january" niveau="month" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
        <item text="February" id="02_february" niveau="month" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
        <item text="March" id="03_march" niveau="month" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
      </item>
      <item text="Spring" id="t2" niveau="trimester" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif">
        <item text="April" id="04_april" niveau="month" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
        <item text="May" id="05_may" niveau="month" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
        <item text="June" id="06_june" niveau="month" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
      </item>
      <item text="Summer" id="t3" niveau="trimester" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif">
        <item text="July" id="07_july" niveau="month" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
        <item text="August" id="08_august" niveau="month" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
        <item text="September" id="09_september" niveau="month" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
      </item>
      <item text="Fall" id="t4" niveau="trimester" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif">
        <item text="October" id="10_october" niveau="month" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
        <item text="November" id="11_november" niveau="month" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
        <item text="December" id="12_december" niveau="month" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
      </item>
    </item>
  </item>
  <item text="Hour range" id="hour_range" niveau="all" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif">
    <item text="00h-03h" id="00h-03h" niveau="hour_range" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
    <item text="03h-06h" id="03h-06h" niveau="hour_range" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
    <item text="06h-09h" id="06h-09h" niveau="hour_range" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
    <item text="09h-12h" id="09h-12h" niveau="hour_range" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
    <item text="12h-15h" id="12h-15h" niveau="hour_range" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
    <item text="15h-18h" id="15h-18h" niveau="hour_range" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
    <item text="18h-21h" id="18h-21h" niveau="hour_range" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
    <item text="21h-00h" id="21h-00h" niveau="hour_range" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
  </item>
</tree>
```

```

</item>
<item text="Day of the week" id="week_day" niveau="all" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif">
  <item text="Monday" id="1" niveau="week_day" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
  <item text="Tuesday" id="2" niveau="week_day" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
  <item text="Wednesday" id="3" niveau="week_day" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
  <item text="Thursday" id="4" niveau="week_day" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
  <item text="Friday" id="5" niveau="week_day" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
  <item text="Saturday" id="6" niveau="week_day" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
  <item text="Sunday" id="0" niveau="week_day" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
</item>
<item text="Crime type" id="crime_group" niveau="all" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif">
  <item text="ASSISTING THE PUBLIC" id="ASSISTING THE PUBLIC" niveau="crime_group" im0="folderClosed.gif" im1="folderOpen.gif"
  im2="folderClosed.gif">
    <item text="MENTAL HEALTH" id="MENTAL HEALTH" niveau="crime_type1" im0="folderClosed.gif" im1="folderOpen.gif"
    im2="folderClosed.gif"/>
    <item text="PERSON DOWN/INJURY" id="PERSON DOWN/INJURY" niveau="crime_type1" im0="folderClosed.gif"
    im1="folderOpen.gif" im2="folderClosed.gif"/>
    <item text="PERSONS - LOST, FOUND, MISSING" id="PERSONS - LOST, FOUND, MISSING" niveau="crime_type1"
    im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
  </item>
  <item text="CRIMES AGAINST PERSONS" id="CRIMES AGAINST PERSONS" niveau="crime_group" im0="folderClosed.gif"
  im1="folderOpen.gif" im2="folderClosed.gif">
    <item text="ASSAULTS" id="ASSAULTS" niveau="crime_type1" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
    <item text="DRIVE BY (NO INJURY)" id="DRIVE BY (NO INJURY)" niveau="crime_type1" im0="folderClosed.gif" im1="folderOpen.gif"
    im2="folderClosed.gif"/>
    <item text="HOMICIDE" id="HOMICIDE" niveau="crime_type1" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
    <item text="ROBBERY" id="ROBBERY" niveau="crime_type1" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
    <item text="THREATS, HARASSMENT" id="THREATS, HARASSMENT" niveau="crime_type1" im0="folderClosed.gif"
    im1="folderOpen.gif" im2="folderClosed.gif"/>
  </item>
  <item text="DRUGS AND VICE" id="DRUGS AND VICE" niveau="crime_group" im0="folderClosed.gif" im1="folderOpen.gif"
  im2="folderClosed.gif">
    <item text="LIQUOR VIOLATIONS" id="LIQUOR VIOLATIONS" niveau="crime_type1" im0="folderClosed.gif" im1="folderOpen.gif"
    im2="folderClosed.gif"/>
    <item text="NARCOTICS COMPLAINTS" id="NARCOTICS COMPLAINTS" niveau="crime_type1" im0="folderClosed.gif"
    im1="folderOpen.gif" im2="folderClosed.gif"/>
    <item text="OTHER VICE" id="OTHER VICE" niveau="crime_type1" im0="folderClosed.gif" im1="folderOpen.gif"
    im2="folderClosed.gif"/>
  </item>
</item>

```

Annexe 1. Fichiers XML de définition des hypercubes raster

```
<item text="PROSTITUTION" id="PROSTITUTION" niveau="crime_type1" im0="folderClosed.gif" im1="folderOpen.gif"
im2="folderClosed.gif"/>
</item>
<item text="PROPERTY CRIME/THEFT" id="PROPERTY CRIME/THEFT" niveau="crime_group" im0="folderClosed.gif" im1="folderOpen.gif"
im2="folderClosed.gif">
  <item text="AUTO THEFTS" id="AUTO THEFTS" niveau="crime_type1" im0="folderClosed.gif" im1="folderOpen.gif"
im2="folderClosed.gif"/>
  <item text="BIKE" id="BIKE" niveau="crime_type1" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
  <item text="BURGLARY" id="BURGLARY" niveau="crime_type1" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
  <item text="CAR PROWL" id="CAR PROWL" niveau="crime_type1" im0="folderClosed.gif" im1="folderOpen.gif"
im2="folderClosed.gif"/>
  <item text="FRAUD CALLS" id="FRAUD CALLS" niveau="crime_type1" im0="folderClosed.gif" im1="folderOpen.gif"
im2="folderClosed.gif"/>
  <item text="OTHER PROPERTY" id="OTHER PROPERTY" niveau="crime_type1" im0="folderClosed.gif" im1="folderOpen.gif"
im2="folderClosed.gif"/>
  <item text="PROPERTY DAMAGE" id="PROPERTY DAMAGE" niveau="crime_type1" im0="folderClosed.gif" im1="folderOpen.gif"
im2="folderClosed.gif"/>
  <item text="PROPERTY - MISSING, FOUND" id="PROPERTY - MISSING, FOUND" niveau="crime_type1" im0="folderClosed.gif"
im1="folderOpen.gif" im2="folderClosed.gif"/>
  <item text="SHOPLIFTING" id="SHOPLIFTING" niveau="crime_type1" im0="folderClosed.gif" im1="folderOpen.gif"
im2="folderClosed.gif"/>
</item>
<item text="TRANSPORTATION" id="TRANSPORTATION" niveau="crime_group" im0="folderClosed.gif" im1="folderOpen.gif"
im2="folderClosed.gif">
  <item text="ACCIDENT INVESTIGATION" id="ACCIDENT INVESTIGATION" niveau="crime_type1" im0="folderClosed.gif"
im1="folderOpen.gif" im2="folderClosed.gif"/>
  <item text="HARBOR CALLS" id="HARBOR CALLS" niveau="crime_type1" im0="folderClosed.gif" im1="folderOpen.gif"
im2="folderClosed.gif"/>
  <item text="LEWD CONDUCT" id="LEWD CONDUCT" niveau="crime_type1" im0="folderClosed.gif" im1="folderOpen.gif"
im2="folderClosed.gif"/>
  <item text="TRAFFIC RELATED CALLS" id="TRAFFIC RELATED CALLS" niveau="crime_type1" im0="folderClosed.gif"
im1="folderOpen.gif" im2="folderClosed.gif"/>
</item>
</item>
</tree>
```

Moon.xml

[illegible]

Annexe 1. Fichiers XML de définition des hypercubes raster

[illegible]

[illegible]

Annexe 1. Fichiers XML de définition des hypercubes raster

```
</item>
<item text="t10" id="10" niveau="time10" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif">
  <item text="91" id="091" niveau="time100" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
  <item text="92" id="092" niveau="time100" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
  <item text="93" id="093" niveau="time100" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
  <item text="94" id="094" niveau="time100" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
  <item text="95" id="095" niveau="time100" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
  <item text="96" id="096" niveau="time100" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
  <item text="97" id="097" niveau="time100" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
  <item text="98" id="098" niveau="time100" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
  <item text="99" id="099" niveau="time100" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
  <item text="100" id="100" niveau="time100" im0="folderClosed.gif" im1="folderOpen.gif" im2="folderClosed.gif"/>
</item>
</item>
</tree>
```


Annexe 2. Exemple de fichier map pour la définition WMS du résultat de l'interface cartographique (MapServer)

MAP

```
STATUS ON
imageCOLOR 0 0 0
IMAGETYPE PNG
SIZE 600 600
NAME "density"
PROJECTION
    "init=epsg:27700"
END
```

```
EXTENT -180 -90 180 90
```

WEB

```
METADATA
"wms_title" "Raster_fact"
"wms_onlineresource"
"http://139.165.56.104/cgi-
bin/mapserver?map=/var/www/html/rastercube/map_files/jpk.map&"
"wms_srs" "epsg:27700"
"wms_format" "image/png"
"wms_enable_request" "*"
END
```

END

LAYER

```
NAME density
TYPE raster
STATUS DEFAULT
DATA "PG:host=139.165.56.104 port=5432 dbname='backup'
user='solap' password='!geomatics21' schema='public' table='map_raster' where='rid=14'
mode=1"
METADATA
    "wms_title" "density" ##required
END
PROCESSING "NODATA=0"
PROCESSING "SCALE=0,14"
CLASS
    NAME "stretch"
    EXPRESSION ([pixel] >= 0)
    STYLE
        COLORRANGE 255 255 255 255 0 0
        DATARANGE 0 14
        RANGEITEM "crime"
    END
END
```

END

```
END
  LEGEND
  KEYSIZE 18 10
  LABEL
    TYPE BITMAP
    SIZE MEDIUM
    COLOR 0 0 89
  END
  STATUS ON
END

END
```

Annexe 3. Fonction KDE dans PostGIS

Création de la fonction KDE (langage plpgsql)

```
-- Function: __kde(double precision, raster)

-- DROP FUNCTION __kde(double precision, raster);

CREATE OR REPLACE FUNCTION __kde(bandwidth double precision, rast raster)
  RETURNS raster AS
$BODY$
DECLARE
  xcorner double precision;
  ycorner double precision;
  resolutionx double precision;
  resolutiony double precision;
  width double precision;
  height double precision;
  ynew double precision;
  xnew double precision;
  kde_value double precision;
  kde_matrix double precision[];
  srid integer;
  distance double precision[];
  point_value integer[];
  length integer;
  kde_term double precision;
  query character varying;
  constant double precision;

BEGIN
  SELECT ST_UpperLeftX(rast) INTO xcorner;
  SELECT ST_UpperLeftY(rast) INTO ycorner;
  SELECT ST_ScaleX(rast) INTO resolutionx;
  SELECT ST_ScaleY(rast) INTO resolutiony;
  SELECT ST_Width(rast) into width;
  SELECT ST_Height(rast) into height;
  SELECT ST_Srid(rast) into srid;

  xcorner=xcorner + resolutionx/2;
  ycorner=ycorner + resolutiony/2;
  constant = 3/(pi()*power(bandwidth, 2))*1000000;
  FOR j in 0..height-1 LOOP
    ynew=ycorner+j*resolutiony;
    FOR i in 0..width-1 LOOP
      xnew=xcorner+i*resolutionx;
      SELECT
        array_agg(
          st_distance(
            st_setsrid(
              st_wkttosql(
                'POINT('||xnew||' '||ynew||')'
              ),
              srid),
          distance,
```

```

        geom)
    ), array_agg(value)
    INTO distance, point_value
    FROM kde_points
    WHERE st_intersects(
        st_buffer(
            st_setsrid(
                st_wkttosql(
                    'POINT('||xnew||' '||ynew||')'
                ), srid
            ), bandwidth
        ), geom
    );

    SELECT array_length(point_value, 1 ) into length;
    kde_value:=0;

    IF length IS NOT NULL THEN
        FOR k in 1..length LOOP
            ----- CHOICE OF KDE FUNCTION
            kde_term          =          point_value[k]*constant*power(1-
power(distance[k]/bandwidth,          2), 2);
            -----
            kde_value:=kde_value+kde_term;
        END LOOP;
    END IF;
    kde_matrix[i]=kde_value;
END LOOP;
SELECT ST_SetValues(rast, 1, 1, j+1, kde_matrix) into rast;
END LOOP;

RETURN rast;
END;
$BODY$
LANGUAGE plpgsql STABLE
COST 100;
ALTER FUNCTION __kde(double precision, raster)
OWNER TO adminuser;

```

Exemple d'utilisation de la fonction KDE (langage SQL)

```
-- Create the table that will contain the Kernel Density Estimation raster
DROP table if exists raster_table;
create table raster_table(
    rid serial,
    rast raster,
    CONSTRAINT raster_table_pkey PRIMARY KEY (rid)
);
-- Create an empty raster
insert into raster_table (rast)
(select st_makeemptyraster(195, 150, 503567.212147483, 200851.873341834, 300, -300, 0, 0, 27700));
-- Add a band to the empty raster
UPDATE raster_table
SET rast = ST_AddBand(rast, 1, '32BF', 0, 0) where rid=1;
-- Create a temporary table with KDE input points. The table has to be called "kde_points". The geometry
column has to be called "geom" and the value of a point has to be called "count"
-- In this case, a point values is the number of data points which are snapped to a grid (10 meters resolution).
Indeed, it makes the KDE operation faster.
-- You can use the point values for a weighted KDE if you want.
-- My data table (crime_point2) is made of crimes with three attributes: crime_type (text), month (date) and
location_proj(geometry)
DROP TABLE IF EXISTS kde_points;
DROP index IF EXISTS kde_points_index;
create table kde_points as(
select count(location_proj) as value, ST_SnapToGrid(location_proj, 10) as geom
from crime_point2
where crime_type='Vehicle crime'
and month='2012-01-01'
group by geom
);
create index kde_points_index on kde_points USING gist(geom);
-- Create the Kernel Density Estimation raster
-- First parameter is the bandwidth
-- Second parameter is the raster column
update raster_table set rast= __kde(1500, rast) where rid=1
```


Annexe 4. Requêtes SQL de l'étude comparative

Requête 1 selon l'approche SOLAP raster :

```
SELECT ST_Union(rast100, 2,'sum') FROM fact_raster
```

Requête 1 selon l'approche SOLAP vectorielle :

```
select count(id_crime), cell100.id, cell100.geom  
from crime_point2, cell100  
where crime_point2.cell100=cell100.id  
group by cell100.id
```

Requête 1 selon l'approche SIG:

```
select cell100.id, count(id_crime) as measure, cell100.geom  
from crime_point2, cell100  
where st_intersects(cell100.geom, crime_point2.location_proj)  
group by cell100.id
```

Requête 2 selon l'approche SOLAP raster :

```
SELECT ST_Union(rast300, 2,'sum') FROM fact_raster
```

Requête 2 selon l'approche SOLAP vectorielle :

```
select cell300.id, count(id_crime) as measure, cell300.geom  
from crime_point2, cell100, cell300  
where crime_point2.cell100=cell100.id  
and cell100.cell300 = cell300.id  
group by cell300.id
```

Requête 2 selon l'approche SIG:

```
select cell300.id, count(id_crime) as measure, cell300.geom  
from crime_point2, cell300  
where st_intersects(cell300.geom, crime_point2.location_proj)  
group by cell300.id
```

Requête 3 selon l'approche SOLAP raster :

```
SELECT ST_Union(rast1000, 2,'sum') FROM fact_raster
```

Requête 3 selon l'approche SOLAP vectorielle :

```
select count(id_crime), cell1000.id, cell1000.geom  
from crime_point2, cell1000  
where crime_point2.cell1000=cell1000.id
```

group by cell1000.id

Requête 3 selon l'approche SIG:

```
select count(id_crime), cell1000.id, cell1000.geom
from crime_point2, cell1000
where st_intersects(cell1000.geom, crime_point2.location_proj)
group by cell1000.id
```

Requête 4 selon l'approche SOLAP raster :

```
SELECT ST_Union(rast100, 2,'sum') FROM fact_raster where crime_type='Burglary'
```

Requête 4 selon l'approche SOLAP vectorielle :

```
select count(id_crime), cell100.id, cell100.geom
from crime_point2, cell100
where crime_point2.cell100=cell100.id
and crime_type='Burglary'
group by cell100.id
```

Requête 4 selon l'approche SIG:

```
select cell100.id, count(id_crime) as measure, cell100.geom
from crime_point2, cell100
where st_intersects(cell100.geom, crime_point2.location_proj)
and crime_type='Burglary'
group by cell100.id
```

Requête 5 selon l'approche SOLAP raster :

```
SELECT ST_Union(rast300, 2,'sum') FROM fact_raster where crime_type='Burglary'
```

Requête 5 selon l'approche SOLAP vectorielle :

```
select cell300.id, count(id_crime) as measure, cell300.geom
from crime_point2, cell100, cell300
where crime_point2.cell100=cell100.id
and crime_type='Burglary'
and cell100.cell300 = cell300.id
group by cell300.id
```

Requête 5 selon l'approche SIG:

```
select cell300.id, count(id_crime) as measure, cell300.geom
from crime_point2, cell300
where st_intersects(cell300.geom, crime_point2.location_proj)
and crime_type='Burglary'
group by cell300.id
```

Requête 6 selon l'approche SOLAP raster :

```
SELECT ST_Union(rast1000, 2,'sum') FROM fact_raster where crime_type='Burglary'
```

Requête 6 selon l'approche SOLAP vectorielle :

```
select count(id_crime), cell1000.id, cell1000.geom
from crime_point2, cell1000
where crime_point2.cell1000=cell1000.id
and crime_type='Burglary'
group by cell1000.id
```

Requête 6 selon l'approche SIG:

```
select cell1000.id, count(id_crime) as measure, cell1000.geom
from crime_point2, cell1000
where st_intersects(cell1000.geom, crime_point2.location_proj)
and crime_type='Burglary'
group by cell1000.id
```

Requête 7 selon l'approche SOLAP raster :

```
SELECT ST_Union(rast100, 2,'sum') FROM fact_raster where crime_type='Burglary'
and month= '2012-01-01'
```

Requête 7 selon l'approche SOLAP vectorielle :

```
select count(id_crime), cell100.id, cell100.geom
from crime_point2, cell100
where crime_point2.cell100=cell100.id
and crime_type='Burglary'
and month='2012-01-01'
group by cell100.id
```

Requête 7 selon l'approche SIG:

```
select cell100.id, count(id_crime) as measure, cell100.geom
from crime_point2, cell100
where st_intersects(cell100.geom, crime_point2.location_proj)
and crime_type='Burglary'
and month='2012-01-01'
group by cell100.id
```

Requête 8 selon l'approche SOLAP raster :

```
SELECT ST_Union(rast300, 2,'sum') FROM fact_raster where crime_type='Burglary'
and month= '2012-01-01'
```

Requête 8 selon l'approche SOLAP vectorielle :

```
select cell300.id, count(id_crime) as measure, cell300.geom
from crime_point2, cell100, cell300
where crime_point2.cell100=cell100.id
and crime_type='Burglary'
and month='2012-01-01'
and cell100.cell300 = cell300.id
group by cell300.id
```

Requête 8 selon l'approche SIG:

```
select cell300.id, count(id_crime) as measure, cell300.geom
from crime_point2, cell300
where st_intersects(cell300.geom, crime_point2.location_proj)
and crime_type='Burglary'
and month='2012-01-01'
group by cell300.id
```

Requête 9 selon l'approche SOLAP raster :

```
SELECT ST_Union(rast1000, 2,'sum') FROM fact_raster where crime_type='Burglary'
and month= '2012-01-01'
```

Requête 9 selon l'approche SOLAP vectorielle :

```
select count(id_crime), cell1000.id, cell1000.geom
from crime_point2, cell1000
where crime_point2.cell1000=cell1000.id
and crime_type='Burglary'
and month='2012-01-01'
group by cell1000.id
```

Requête 9 selon l'approche SIG:

```
select cell1000.id, count(id_crime) as measure, cell1000.geom
from crime_point2, cell1000
where st_intersects(cell1000.geom, crime_point2.location_proj)
and crime_type='Burglary'
and month='2012-01-01'
group by cell1000.id
```

Requête 10 selon l'approche SOLAP raster :

```
SELECT ST_Union(little_rast.st_mapalgebraexpr,'sum')
FROM (SELECT ST_MapAlgebraExpr(fact.rast100, 2, mask.st_asraster, 1, '[rast1.val]*[rast2.val]::double
precision', '32BF', 'INTERSECTION', '0', '0')
from (select rast100 from fact_raster )as fact, (select st_asraster from maskjpk) as mask) as little_rast
```

Requête 10 selon l'approche SOLAP vectorielle :

```
select count(id_crime), cell100.id, cell100.geom
from crime_point2, cell100, geodimension
where crime_point2.cell100=cell100.id
and st_intersects(st_transform(st_collectionextract(geodimension.geom,3),27700), cell100.geom)
group by cell100.id
```

Requête 10 selon l'approche SIG:

```
select cell100.id, count(id_crime) as measure, cell100.geom
from crime_point2, cell100, geodimension
where st_intersects(cell100.geom, crime_point2.location_proj)
and st_intersects(st_transform(st_collectionextract(geodimension.geom,3),27700), cell100.geom)
```

group by cell100.id

Requête 11 selon l'approche SOLAP raster :

```
SELECT ST_Union(little_rast.st_mapalgebraexpr,'sum')
FROM (SELECT ST_MapAlgebraExpr(fact.rast300, 2, mask.st_asraster, 1, '[rast1.val]*[rast2.val]::double
precision', '32BF', 'INTERSECTION', '0', '0')
from (select rast300 from fact_raster )as fact, (select st_asraster from maskjpk) as mask) as little_rast
```

Requête 11 selon l'approche SOLAP vectorielle :

```
select cell300.id, count(id_crime) as measure, cell300.geom
from crime_point2, cell100, cell300, geodimension
where crime_point2.cell100=cell100.id
and cell100.cell300 = cell300.id
and st_intersects(st_transform(st_collectionextract(geodimension.geom,3),27700), cell300.geom)
group by cell300.id
```

Requête 11 selon l'approche SIG:

```
select cell300.id, count(id_crime) as measure, cell300.geom
from crime_point2, cell300, geodimension
where st_intersects(cell300.geom, crime_point2.location_proj)
and st_intersects(st_transform(st_collectionextract(geodimension.geom,3),27700), cell300.geom)
group by cell300.id
```

Requête 12 selon l'approche SOLAP raster :

```
SELECT ST_Union(little_rast.st_mapalgebraexpr,'sum')
FROM (SELECT ST_MapAlgebraExpr(fact.rast1000, 2, mask.st_asraster, 1, '[rast1.val]*[rast2.val]::double
precision', '32BF', 'INTERSECTION', '0', '0')
from (select rast1000 from fact_raster )as fact, (select st_asraster from maskjpk) as mask) as little_rast
```

Requête 12 selon l'approche SOLAP vectorielle :

```
select count(id_crime), cell1000.id, cell1000.geom
from crime_point2, cell1000, geodimension
where crime_point2.cell1000=cell1000.id
and st_intersects(st_transform(st_collectionextract(geodimension.geom,3),27700), cell1000.geom)
group by cell1000.id
```

Requête 12 selon l'approche SIG:

```
select cell1000.id, count(id_crime) as measure, cell1000.geom
from crime_point2, cell1000, geodimension
where st_intersects(cell1000.geom, crime_point2.location_proj)
and st_intersects(st_transform(st_collectionextract(geodimension.geom,3),27700), cell1000.geom)
group by cell1000.id
```

Requête 13 selon l'approche SOLAP raster :

```
SELECT ST_Union(little_rast.st_mapalgebraexpr,'sum')
```

```
FROM (SELECT ST_MapAlgebraExpr(fact.rast100, 2, mask.st_asraster, 1, '[rast1.val]*[rast2.val]::double
precision', '32BF', 'INTERSECTION', '0', '0')
from (select rast100 from fact_raster where crime_type='Burglary') as fact, (select st_asraster from maskjpk)
as mask) as little_rast
```

Requête 13 selon l'approche SOLAP vectorielle :

```
select count(id_crime) as measure, cell100.id, cell100.geom
from crime_point2, cell100, geodimension
where crime_point2.cell100=cell100.id
and st_intersects(st_transform(st_collectionextract(geodimension.geom,3),27700), cell100.geom)
and crime_type='Burglary'
group by cell100.id
```

Requête 13 selon l'approche SIG:

```
select cell100.id, count(id_crime) as measure, cell100.geom
from crime_point2, cell100, geodimension
where st_intersects(cell100.geom, crime_point2.location_proj)
and st_intersects(st_transform(st_collectionextract(geodimension.geom,3),27700), cell100.geom)
and crime_type='Burglary'
group by cell100.id
```

Requête 14 selon l'approche SOLAP raster :

```
SELECT ST_Union(little_rast.st_mapalgebraexpr,'sum')
FROM (SELECT ST_MapAlgebraExpr(fact.rast300, 2, mask.st_asraster, 1, '[rast1.val]*[rast2.val]::double
precision', '32BF', 'INTERSECTION', '0', '0')
from (select rast300 from fact_raster where crime_type='Burglary')as fact, (select st_asraster from maskjpk)
as mask) as little_rast
```

Requête 14 selon l'approche SOLAP vectorielle :

```
select cell300.id, count(id_crime) as measure, cell300.geom
from crime_point2, cell100, cell300, geodimension
where crime_point2.cell100=cell100.id
and cell100.cell300 = cell300.id
and st_intersects(st_transform(st_collectionextract(geodimension.geom,3),27700), cell100.geom)
and crime_type='Burglary'
group by cell300.id
```

Requête 14 selon l'approche SIG:

```
select cell300.id, count(id_crime) as measure, cell300.geom
from crime_point2, cell300, geodimension
where st_intersects(cell300.geom, crime_point2.location_proj)
and st_intersects(st_transform(st_collectionextract(geodimension.geom,3),27700), cell300.geom)
and crime_type='Burglary'
group by cell300.id
```

Requête 15 selon l'approche SOLAP raster :

```
SELECT ST_Union(little_rast.st_mapalgebraexpr,'sum')
FROM (SELECT ST_MapAlgebraExpr(fact.rast1000, 2, mask.st_asraster, 1, '[rast1.val]*[rast2.val]::double
precision', '32BF', 'INTERSECTION', '0', '0')
```

Annexe 4. Requêtes SQL de l'étude comparative

```
from (select rast1000 from fact_raster where crime_type='Burglary')as fact, (select st_asraster from maskjpk)
as mask) as little_rast
```

Requête 15 selon l'approche SOLAP vectorielle :

```
select count(id_crime), cell1000.id, cell1000.geom
from crime_point2, cell1000, geodimension
where crime_point2.cell1000=cell1000.id
and st_intersects(st_transform(st_collectionextract(geodimension.geom,3),27700), cell1000.geom)
and crime_type='Burglary'
group by cell1000.id
```

Requête 15 selon l'approche SIG:

```
select cell1000.id, count(id_crime) as measure, cell1000.geom
from crime_point2, cell1000, geodimension
where st_intersects(cell1000.geom, crime_point2.location_proj)
and st_intersects(st_transform(st_collectionextract(geodimension.geom,3),27700), cell1000.geom)
and crime_type='Burglary'
group by cell1000.id
```

Requête 16 selon l'approche SOLAP raster :

```
SELECT month, (stats).sum
from ( SELECT ST_SummaryStats(rast) As stats, month
from( SELECT ST_Union(little_rast.st_mapalgebraexpr,'sum') as rast, month
FROM (SELECT ST_MapAlgebraExpr(fact.rast100, 2, mask.st_asraster, 1, '[rast1.val]*[rast2.val]::double
precision', '32BF', 'INTERSECTION', '0', '0'), month
from (select rast100, month from fact_raster
)as fact, maskjpk as mask
) as little_rast group by month order by month
) as foo
) as stats
```

Requête 16 selon l'approche SOLAP vectorielle :

```
select month, count(id_crime)
from crime_point2, cell100, geodimension
where crime_point2.cell100=cell100.id
and st_intersects(st_transform(st_collectionextract(geodimension.geom,3),27700), cell100.geom)
group by month order by month
```

Requête 16 selon l'approche SIG:

```
select count(id_crime) as measure, month
from crime_point2, geodimension
where
st_intersects(st_transform(st_collectionextract(geodimension.geom,3),27700), location_proj)
group by month order by month
```

Requête 17 selon l'approche SOLAP raster :

```
SELECT month, (stats).sum
from ( SELECT ST_SummaryStats(rast) As stats, month
```

```

from( SELECT ST_Union(little_rast.st_mapalgebraexpr,'sum') as rast, month
FROM (SELECT ST_MapAlgebraExpr(fact.rast300, 2, mask.st_asraster, 1, '[rast1.val]*[rast2.val]::double
precision', '32BF', 'INTERSECTION', '0', '0'), month
from (select rast300, month from fact_raster
)as fact, maskjpk as mask
) as little_rast group by month order by month
) as foo
) as stats

```

Requête 17 selon l'approche SOLAP vectorielle :

```

select month, count(id_crime) as measure
from crime_point2, cell100, cell300, geodimension
where crime_point2.cell100=cell100.id
and cell100.cell300 = cell300.id
and st_intersects(st_transform(st_collectionextract(geodimension.geom,3),27700), cell300.geom)
group by month order by month

```

Requête 17 selon l'approche SIG:

```

select count(id_crime) as measure, month
from crime_point2, geodimension
where
st_intersects(st_transform(st_collectionextract(geodimension.geom,3),27700), location_proj)
group by month order by month

```

Requête 18 selon l'approche SOLAP raster :

```

SELECT month, (stats).sum
from ( SELECT ST_SummaryStats(rast) As stats, month
from( SELECT ST_Union(little_rast.st_mapalgebraexpr,'sum') as rast, month
FROM (SELECT ST_MapAlgebraExpr(fact.rast1000, 2, mask.st_asraster, 1, '[rast1.val]*[rast2.val]::double
precision', '32BF', 'INTERSECTION', '0', '0'), month
from (select rast1000, month from fact_raster
)as fact, maskjpk as mask
) as little_rast group by month order by month
) as foo
) as stats

```

Requête 18 selon l'approche SOLAP vectorielle :

```

select month, count(id_crime)
from crime_point2, cell1000, geodimension
where crime_point2.cell1000=cell1000.id
and st_intersects(st_transform(st_collectionextract(geodimension.geom,3),27700), cell1000.geom)
group by month order by month

```

Requête 18 selon l'approche SIG:

```

select count(id_crime) as measure, month
from crime_point2, geodimension
where
st_intersects(st_transform(st_collectionextract(geodimension.geom,3),27700), location_proj)
group by month order by month

```

Requête 19 selon l'approche SOLAP raster :

```
SELECT month, (stats).sum
from ( SELECT ST_SummaryStats(rast) As stats, month
from( SELECT ST_Union(little_rast.st_mapalgebraexpr,'sum') as rast, month
FROM (SELECT ST_MapAlgebraExpr(fact.rast100, 2, mask.st_asraster, 1, '[rast1.val]*[rast2.val]::double
precision', '32BF', 'INTERSECTION', '0', '0'), month
from (select rast100, month from fact_raster where crime_type='Burglary'
)as fact, maskjpk as mask
) as little_rast group by month order by month
) as foo
) as stats
```

Requête 19 selon l'approche SOLAP vectorielle :

```
select month, count(id_crime)
from crime_point2, cell100, geodimension
where crime_point2.cell100=cell100.id
and st_intersects(st_transform(st_collectionextract(geodimension.geom,3),27700), cell100.geom)
and crime_type='Burglary'
group by month order by month
```

Requête 19 selon l'approche SIG:

```
select count(id_crime) as measure, month
from crime_point2, geodimension
where
st_intersects(st_transform(st_collectionextract(geodimension.geom,3),27700), location_proj)
and crime_type='Burglary'
group by month order by month
```

Requête 20 selon l'approche SOLAP raster :

```
SELECT month, (stats).sum
from ( SELECT ST_SummaryStats(rast) As stats, month
from( SELECT ST_Union(little_rast.st_mapalgebraexpr,'sum') as rast, month
FROM (SELECT ST_MapAlgebraExpr(fact.rast300, 2, mask.st_asraster, 1, '[rast1.val]*[rast2.val]::double
precision', '32BF', 'INTERSECTION', '0', '0'), month
from (select rast300, month from fact_raster where crime_type='Burglary'
)as fact, maskjpk as mask
) as little_rast group by month order by month
) as foo
) as stats
```

Requête 20 selon l'approche SOLAP vectorielle :

```
select month, count(id_crime) as measure
from crime_point2, cell100, cell300, geodimension
where crime_point2.cell100=cell100.id
and cell100.cell300 = cell300.id
and st_intersects(st_transform(st_collectionextract(geodimension.geom,3),27700), cell300.geom)
and crime_type='Burglary'
group by month order by month
```

Requête 20 selon l'approche SIG:

```
select count(id_crime) as measure, month
from crime_point2, geodimension
where
st_intersects(st_transform(st_collectionextract(geodimension.geom,3),27700), location_proj)
and crime_type='Burglary'
group by month order by month
```

Requête 21 selon l'approche SOLAP raster :

```
SELECT month, (stats).sum
from ( SELECT ST_SummaryStats(rast) As stats, month
from( SELECT ST_Union(little_rast.st_mapalgebraexpr,'sum') as rast, month
FROM (SELECT ST_MapAlgebraExpr(fact.rast1000, 2, mask.st_asraster, 1, '[rast1.val]*[rast2.val]::double
precision', '32BF', 'INTERSECTION', '0', '0'), month
from (select rast1000, month from fact_raster where crime_type='Burglary'
)as fact, maskjpk as mask
) as little_rast group by month order by month
) as foo
) as stats
```

Requête 21 selon l'approche SOLAP vectorielle :

```
select month, count(id_crime)
from crime_point2, cell1000, geodimension
where crime_point2.cell1000=cell1000.id
and st_intersects(st_transform(st_collectionextract(geodimension.geom,3),27700), cell1000.geom)
and crime_type='Burglary'
group by month order by month
```

Requête 21 selon l'approche SIG:

```
select count(id_crime) as measure, month
from crime_point2, geodimension
where
st_intersects(st_transform(st_collectionextract(geodimension.geom,3),27700), location_proj)
and crime_type='Burglary'
group by month order by month
```

Requête 22 selon l'approche SOLAP raster :

```
SELECT month, crime_type, (stats).sum
from ( SELECT ST_SummaryStats(rast) As stats, month , crime_type
from( select st_union(rast100,2,'sum') as rast, month, crime_type
from fact_raster group by month, crime_type order by month, crime_type)
as result )
as foo
```

Requête 22 selon l'approche SOLAP vectorielle :

```
select month, crime_type, count(id_crime)
from crime_point2
group by month, crime_type order by month, crime_type
```

Requête 22 selon l'approche SIG:

```
select month, crime_type, count(id_crime)
from crime_point2
group by month, crime_type order by month, crime_type
```

Requête 23 selon l'approche SOLAP raster :

```
SELECT month, crime_type, (stats).sum
from ( SELECT ST_SummaryStats(rast) As stats, month , crime_type
from( select st_union(rast300,2,'sum') as rast, month, crime_type
from fact_raster group by month, crime_type order by month, crime_type)
as result )
as foo
```

Requête 23 selon l'approche SOLAP vectorielle :

```
select month, crime_type, count(id_crime)
from crime_point2
group by month, crime_type order by month, crime_type
```

Requête 23 selon l'approche SIG:

```
select month, crime_type, count(id_crime)
from crime_point2
group by month, crime_type order by month, crime_type
```

Requête 24 selon l'approche SOLAP raster :

```
SELECT month, crime_type, (stats).sum
from ( SELECT ST_SummaryStats(rast) As stats, month , crime_type
from( select st_union(rast1000,2,'sum') as rast, month, crime_type
from fact_raster group by month, crime_type order by month, crime_type)
as result )
as foo
```

Requête 24 selon l'approche SOLAP vectorielle :

```
select month, crime_type, count(id_crime)
from crime_point2
group by month, crime_type order by month, crime_type
```

Requête 24 selon l'approche SIG:

```
select month, crime_type, count(id_crime)
from crime_point2
group by month, crime_type order by month, crime_type
```

Requête 25 selon l'approche SOLAP raster :

```
SELECT month, crime_type, (stats).sum
from ( SELECT ST_SummaryStats(rast) As stats, month, crime_type
from( SELECT ST_Union(little_rast.st_mapalgebraexpr,'sum') as rast, month, crime_type
```

```

FROM (SELECT ST_MapAlgebraExpr(fact.rast100, 2, mask.st_asraster, 1, '[rast1.val]*[rast2.val]::double
precision', '32BF', 'INTERSECTION', '0', '0'), month, crime_type
from (select rast100, month, crime_type from fact_raster
)as fact, maskjpk as mask
) as little_rast group by month, crime_type order by month, crime_type
) as foo
) as stats

```

Requête 25 selon l'approche SOLAP vectorielle :

```

select month, crime_type, count(id_crime) as measure
from crime_point2, cell100, geodimension
where crime_point2.cell100=cell100.id
and st_intersects(st_transform(st_collectionextract(geodimension.geom,3),27700), cell100.geom)
group by month, crime_type order by month

```

Requête 25 selon l'approche SIG:

```

select count(id_crime) as measure, month, crime_type
from crime_point2, geodimension
where
st_intersects(st_transform(st_collectionextract(geodimension.geom,3),27700), location_proj)
group by month, crime_type order by month, crime_type

```

Requête 26 selon l'approche SOLAP raster :

```

SELECT month, crime_type, (stats).sum
from ( SELECT ST_SummaryStats(rast) As stats, month, crime_type
from( SELECT ST_Union(little_rast.st_mapalgebraexpr,'sum') as rast, month, crime_type
FROM (SELECT ST_MapAlgebraExpr(fact.rast300, 2, mask.st_asraster, 1, '[rast1.val]*[rast2.val]::double
precision', '32BF', 'INTERSECTION', '0', '0'), month, crime_type
from (select rast300, month, crime_type from fact_raster
)as fact, maskjpk as mask
) as little_rast group by month, crime_type order by month, crime_type
) as foo
) as stats

```

Requête 26 selon l'approche SOLAP vectorielle :

```

select month, crime_type, count(id_crime) as measure
from crime_point2, cell100, cell300, geodimension
where crime_point2.cell100=cell100.id
and cell100.cell300 = cell300.id
and st_intersects(st_transform(st_collectionextract(geodimension.geom,3),27700), cell300.geom)
group by month, crime_type order by month

```

Requête 26 selon l'approche SIG:

```

select count(id_crime) as measure, month, crime_type
from crime_point2, geodimension
where
st_intersects(st_transform(st_collectionextract(geodimension.geom,3),27700), location_proj)
group by month, crime_type order by month, crime_type

```

Requête 27 selon l'approche SOLAP raster :

Annexe 4. Requêtes SQL de l'étude comparative

```
SELECT month, crime_type, (stats).sum
from ( SELECT ST_SummaryStats(rast) As stats, month, crime_type
from( SELECT ST_Union(little_rast.st_mapalgebraexpr,'sum') as rast, month, crime_type
FROM (SELECT ST_MapAlgebraExpr(fact.rast1000, 2, mask.st_asraster, 1, '[rast1.val]*[rast2.val]::double
precision', '32BF', 'INTERSECTION', '0', '0'), month, crime_type
from (select rast1000, month, crime_type from fact_raster
)as fact, maskjpk as mask
) as little_rast group by month, crime_type order by month, crime_type
) as foo
) as stats
```

Requête 27 selon l'approche SOLAP vectorielle :

```
select month, crime_type, count(id_crime)
from crime_point2, cell1000, geodimension
where crime_point2.cell1000=cell1000.id
and st_intersects(st_transform(st_collectionextract(geodimension.geom,3),27700), cell1000.geom)
group by month, crime_type order by month, crime_type
```

Requête 27 selon l'approche SIG:

```
select count(id_crime) as measure, month, crime_type
from crime_point2, geodimension
where
st_intersects(st_transform(st_collectionextract(geodimension.geom,3),27700), location_proj)
group by month, crime_type order by month, crime_type
```